Contents lists available at ScienceDirect

Information Fusion

journal homepage: www.elsevier.com/locate/inffus

Full length article

NAPS Fusion: A framework to overcome experimental data limitations to predict human performance and cognitive task outcomes

Nicholas J. Napoli [a,d,*], Chad L. Stephens [b], Kellie D. Kennedy [b], Laura E. Barnes [c], Ezequiel Juarez Garcia [a,d], Angela R. Harrivel [b]

[a] *Human Informatics and Predictive Performance Optimization Laboratory, Electrical and Computer Engineering, University of Florida, Gainesville, FL, 32611, USA*
[b] *NASA Langley Research Center, Hampton, VA 23681, USA*
[c] *Systems and Information Engineering, University of Virginia, Charlottesville, VA 22904, USA*
[d] *National Institute of Aerospace, Hampton, VA 23666, USA*

ARTICLE INFO

Keywords:
Sensor fusion
Human performance prediction
Experimental data variability
Small experimental sample size
Uncertain ground truth
Uncertainty modeling

ABSTRACT

In the area of human performance and cognitive research, machine learning (ML) problems become increasingly complex due to limitations in the experimental design, resulting in the development of poor predictive models. More specifically, experimental study designs produce very few data instances, have large class imbalances and conflicting ground truth labels, and generate wide data sets due to the diverse amount of sensors. From an ML perspective these problems are further exacerbated in anomaly detection cases where class imbalances occur and there are almost always more features than samples. Typically, dimensionality reduction methods (e.g., PCA, autoencoders) are utilized to handle these issues from wide data sets. However, these dimensionality reduction methods do not always map to a lower dimensional space appropriately, and they capture noise or irrelevant information. In addition, when new sensor modalities are incorporated, the entire ML paradigm has to be remodeled because of new dependencies introduced by the new information. Remodeling these ML paradigms is time-consuming and costly due to lack of modularity in the paradigm design, which is not ideal. Furthermore, human performance research experiments, at times, creates ambiguous class labels because the ground truth data cannot be agreed upon by subject-matter experts annotations, making ML paradigm nearly impossible to model.

This work pulls insights from Dempster–Shafer theory (DST), stacking of ML models, and bagging to address uncertainty and ignorance for multi-classification ML problems caused by ambiguous ground truth, low samples, subject-to-subject variability, class imbalances, and wide data sets. Based on these insights, we propose a probabilistic model fusion approach, Naive Adaptive Probabilistic Sensor (NAPS), which combines ML paradigms built around bagging algorithms to overcome these experimental data concerns while maintaining a modular design for future sensor (new feature integration) and conflicting ground truth data. We demonstrate significant overall performance improvements using NAPS (an accuracy of 95.29%) in detecting human task errors (a four class problem) caused by impaired cognitive states and a negligible drop in performance with the case of ambiguous ground truth labels (an accuracy of 93.93%), when compared to other methodologies (an accuracy of 64.91%). This work potentially sets the foundation for other human-centric modeling systems that rely on human state prediction modeling.

1. Introduction

Human–computer interaction research on engineering models of human behavior (i.e., predictive human performance modeling and human cognitive modeling) enable developing enhanced human–machine interfaces, such as biomedical informatics at the bedside in health care settings or human and autonomous systems teaming in aeronautics contexts. A challenge when designing these systems is understanding the state of human operators and their task performance. This understanding can be derived from measures of physiological states, human

[*] Corresponding author at: Human Informatics and Predictive Performance Optimization Laboratory, Electrical and Computer Engineering, University of Florida, Gainesville, FL, 32611, USA.

E-mail addresses: n.napoli@ufl.edu (N.J. Napoli), chad.l.stephens@nasa.gov (C.L. Stephens), kellie.d.kennedy@nasa.gov (K.D. Kennedy), lb3dp@virginia.edu (L.E. Barnes), ejuarezgarcia@ufl.edu (E. Juarez Garcia), angela.r.harrivel@nasa.gov (A.R. Harrivel).

https://doi.org/10.1016/j.inffus.2022.09.016
Received 31 January 2022; Received in revised form 17 September 2022; Accepted 20 September 2022
Available online 27 September 2022

emotions, cognitive states, and errors in performance. The ability to understand the human from the perspective of the system they are linked to can aid task-related activities (e.g., in semi-autonomous vehicles) or provide the required decision support for the system (e.g., intelligent health care systems). In order for a machine to directly adjust and assist the human, we must be able to predict the changes within the human for the machine to intervene and provide decision support. This research proposes a new technology, Naive Adaptive Probabilistic Sensor (NAPS) fusion, to predict cognitive and performance errors using a predictive machine learning (ML) fusion approach that quantifies uncertainty and accounts for ignorance within the model and between ML models.

Prior Work. Under the umbrella of human performance and cognitive predictive research, feasible approaches are needed for highly granular, longitudinal studies to evaluate people in open environments performing everyday tasks. Thus, real-time monitoring of multi-modal psychophysiological, movement and behavioral data is necessary to capture subtle changes in detecting human activities, human states, cognitive states, and other related human states [1–5]. Although analysis of single modality data can predict a specific human state (e.g., cognitive state, psychological state) using traditional statistical methods [6,7], this approach is insufficient for multi-class detection problems and overall lacks the necessary dimensionality to predict multiple classes [8,9]. More specifically, this approach is limited because different physiological and behavioral changes are necessary to detect patterns of various combinations of physiological subsystems [9,10], such as electrocardiograms (ECG) combined with electroencephalogram (EEG) to incorporate the cardiac and neurological systems together. This single modality concept of detecting a human's state would be analogous of trying to triangulate someone's GPS locations using only one satellite. Multiple satellites (i.e., sensors) are required to obtain the person's X, Y, and Z coordinates to obtain their position in space. Why would predicting a human's performance, a more complex analytical problem, be anything less? In real open-world environments, human performance is multidimensional, requiring insight from numerous physiological subsystems, where each subsystem provides additional discriminants independent of other systems about the state of the human.

Challenges. For human–machine interaction, large amounts of annotated data are required in order to properly capture the variance of multiple cognitive changes, human state changes, and subject variability for accurate classification. However, sometimes large-scale data collection on humans is just not possible due to cost, access to a specific population (e.g., rare diseases, a subject's experience level), and time. For example, within the aerospace domain and human performance monitoring, we require: (1) a specialized median to recruit trained pilots for the study; (2) potentially one or multiple full days of data collection for a single human subject's data, (3) potentially multiple medical doctors, technicians, and researchers to oversee the study to maintain data quality and ensure the subjects' safety; (4) flight simulation or real flight time allocation. Collecting a single subject's data can easily cost thousands to tens of thousands of dollars. Thus, building a comprehensive data set for predicting human performance can be severely constrained, which creates complex ML problems that require numerous modalities of streaming sensors (e.g., electrocardiogram, electroencephalogram, etc.) in order to capture systemic psychophysiological changes [10]. These problems are further exacerbated since the quantity of tasks performed in an experiment is also limited by the number of "dependent" samples a single subject can produce, producing an anomaly detection ML problem (e.g., rare outcomes to performance tasks). This is where class imbalances are introduced into the ML data set, and there are more features collected than there are samples, which is commonly referred to as a wide data set. These sensors can create hundreds of features (predictive variables) that often outnumber the number of tasks performed (response variables) in an experiment. The more modalities used to gain knowledge about underlying cognitive states, the more features we obtain, but task-related sample sizes can still limit the usable information. Furthermore, in some cognitive experimental designs, subject-matter experts tend to disagree with data annotations, causing ambiguity for the ground truth annotated data sets [11]. All of these design problems caused by uncertainty are challenging to address due to a lack of training data instances, class imbalances, wide data sets, subject-to-subject variability, and the inability to modularly scale to new sensors from previous ML paradigms. Collectively, these issues place uncertainty on the optimal decision boundary for the ML model and limit its predictive power.

Insights. Dempster–Shafer theory (DST) is a framework equipped to deal with little to no prior knowledge, ignorance, and uncertainty [12,13]. DST was developed for its ability to handle imperfect data in an effective and more intuitive manner [14]. DST has three major caveats when compared to probability theory but can still be considered a generalization of probability [15]. The first main caveat between the two methods is that Bayesian approaches assume that the distribution, otherwise known as probability mass function (p.m.f), is fully defined. Therefore, Bayesian approaches conform strictly to axioms of probability [16,17]. DST relaxes the axiom of additivity by stating that if evidence is not provided or there is conflict, the "support" is assigned to the full set of propositions as uncertainty. Thus, DST handles and quantifies uncertainty for distributions that are not fully defined due to incomplete information. The second caveat is that probability theory only assigns probabilities to "singletons" (e.g., with sample space $\{X, Y, Z\}$, $\{X\}$, $\{Y\}$, $\{Z\}$). In order to determine probabilities associated with other propositions, we examine the union of the probabilities. DST, on the other hand, allows "supports" to be assigned to the complete power set of possibilities, meaning you can set support to various combinations within the set (i.e., doubleton, $\{X, Y\}$). Thus, DST allows us to model ignorance by how we set "supports" to propositions because we are uncertain of which singleton proposition to support. Thirdly, DST approaches are conveniently designed to combine Bodies of Evidence (BoE) through fusion paradigms such as Dempster's Combination Rule (DCR) [18,19]. These BoE act as independent sources of information (i.e., a sensor model) but are combined to update the set of "supports" for the propositions and uncertainty.

Although our approach does not need to be restricted to applying DST, the core concepts of implementing similar frameworks can potentially address the challenges previously discussed. Despite DST's designed to handle uncertainty, the three major parts of the framework that can potentially overcome these challenges are: (1) the concept of combining independent sources of information (i.e., combining multiple models) (2) the concept of setting support to various combinations of the proposition (i.e., augmenting the response variable).

First, this concept of fusing information from multiple BoE together allows us to combine multiple subspaces of information in order to capture the full feature space of the data. This can be achieved by making a single BoE represent a sensor or a small subset of features within the vector space. We can simply build numerous small models associated with a sensor or subset of sensors that are then fused together to expand to a larger feature space. Thus, smaller models have fewer parameters to approximate and fit against, allowing us to reduce our uncertainty within the model's decision boundary. When these smaller ML models are fused together around a DS Framework (e.g., a framework that can handle uncertainty), we can then overcome these constraints caused by high dimensionality and small sample sizes by indirectly loosing these constraints that are directly placed on classical ML modeling approaches. This is the foundation and paramount concept for this proposed framework and enables us to avoid the application of dimensionality reductions to our data. These dimensionality reductions methods potentially fail to capture relevant information caused by the low amount of samples and class imbalances in the data, where all the data is projected in a generalized fashion. This approach adapts and weights each sample independently depending on the ML model's fit. This ability to adapt to each sample and the utility of fusing smaller ML models together also allows the addition of other BoE (i.e., a new sensor modality) to be included in the detection paradigm in later iterative designs for increased modularity.

Secondly, the concept of the framework allows the bodies of evidence (BoE) to avoid being constrained to specifically support any single hypothesis (i.e., fuzzy classification). In real-world cognitive performance problems, subject-matter experts do not always agree on the same cognitive state label. This disagreement of the class label may be attributed to the poor inter-rater reliability, a mixture of cognitive states, or a transition to a new cognitive state. Our framework allows the paradigm to give support to a label that contains multiple classes as a single proposition (i.e., a doubleton, $\{X, Y\}$). This allows us to account for ignorance within the data. These labels with multiple classes essentially merge both response variables together and their data. Thus, the most important takeaway is that we can increase sample size and simplify the model's classification when we merge a response variable together.

To explain NAPS Fusion in detail, the rest of this paper is broken down as follows. In Section 2, we go over the human experimental study design which generated the data for the framework training and validation. This is followed by Section 3 in which we begin constructing the data structures required by the framework. In this section, we also discuss the physiological features extracted from the cognitive study and rely on them to help explain the framework's data structures. Section 4 introduces the more rigorous Dempster–Shafer concepts to elucidate the remaining components of the framework. A lengthy example on mass assignment is given in Section 5 to finalize our explanation of NAPS Fusion. Finally, Section 6 provides our results and discussion.

Contributions. In the aforementioned sections, we develop a new sensor fusion framework named NAPS Fusion to overcome predictive modeling limited limitations due to deficient experimental data. The framework was validated on a multiclass experiment on cognitive state impairment to demonstrate its ability to handle data with class imbalances, ambiguous labels, and few samples. The development of the NAPS Fusion paradigm showed a significant improvement in classification performance compared to other baseline machine learning methods, including deep learning.

2. Experimental design and physiological features

The primary goal of the study was to verify and validate the cognitive capacity of human subjects undergoing normobaric hypoxia induction. Symptoms of hypoxia are shown to cause cognitive impairment that can lead to lapses of attention, loss of situational awareness in operational contexts, temporary mental deficits, and even complete incapacitation, all of which threaten safety of flight. The potentially disastrous consequences of hypoxia in aviation underscore the need for a robust and flexible Human–Machine interaction framework to help understand and prevent negative aviation outcomes [20,21]. Thus, this work provides a strong ML design case for predicting task performance errors caused by cognitive impairments during normal and hypoxic conditions.

Data were collected by a research team at NASA LaRC who subjected 56 volunteers with current hypoxia training certificates to normobaric hypoxia to study the impact on aircraft pilot performance. The data set was later reduced down to 49 subjects due to experimental obstacles and data attrition. Subjects completed informed consent documentation and then were briefed on the operation of the Environics, Inc. Reduced Oxygen Breathing Device (ROBD-2) and connected to physiological recording equipment. In the study, pilots were administered the Multi-Attribute Task Battery (MATB) task, a computer-based cognitive task designed to evaluate operator performance and workload that mimic typical flight tasks. The subjects completed training sessions for the experimental MATB task and sat quietly breathing room air while wearing masks to establish a physiological baseline, shown in Fig. 1. Subjects performed the MATB task three times under the following conditions: (1) breathing room air while wearing a mask; (2) breathing sea level gas mixture through a mask; and (3) breathing 15,000 ft gas mixture through a mask. Between MATB sessions, they recovered from hypoxia exposure by breathing 100% O2 for two minutes following 15,000 ft exposure, and room air was provided for the other sessions. The subjects completed a self-reported workload measure (NASA-Task Load Index, NASA-TLX) after each trial. After completing all trials, subjects were debriefed regarding the study purpose. Data from the NASA study was used to investigate the relationship between pilot physiology and performance under both hypoxic and non-hypoxic conditions. Subjects in the study experienced simulated altitudes of sea level (21% O2) and 15,000 ft (11.2% O2) induced by the ROBD-2.

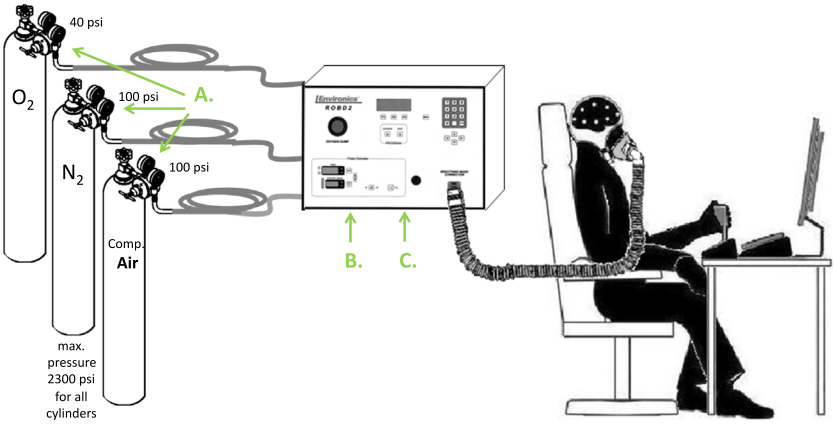
During all trials, multiple bio-sensors collected pilot physiological responses, including electrocardiogram (ECG), electroencephalogram (EEG), electrical dermal activity (EDA), oxygen concentration (O2), oxygen saturation (O2Sat), photoplethysmogram (PPG), and respiration. Since hypoxia has been found to cause cognitive and psychomotor deficits [22,23], we expected induced hypoxia to affect a pilot's ability to perform these tasks. Machine learning and Cognitive modeling designed around the MATB performance can be utilized to capture the behavioral impacts of changes in workload, operator stress level, or fatigue levels caused by hypoxia and characterizes the high-level strategies engaged during continuous multitasking [21,24,25].

2.1. Multi-attribute Task Battery (MATB)

The Multi-Attribute Task Battery (MATB) was used to provide important insight into the applied effects of performance. MATB was developed in 1990 as a test designed to evaluate operator performance and workload via a set of aviation-related tasks [26]. Tasks consist of monitoring, tracking, communication, and resource management, as demonstrated in Fig. 2. The three tasks imposed on pilots using the MATB were tracking, resource management, and communications. The tracking is located in the upper-middle window and requires the test subject to keep the circular target in the center of the window using a joystick. This task is a compensatory task, thus increased reaction time resulting from a hypoxic state could affect a subject's ability to compensate or cause them to overcompensate. Resource management requires subjects to maintain fuel tanks at a level of 2500 units each, which can be achieved by transferring fuel from tank to tank. However, since hypoxia can cause impaired mental arithmetic and decision-making skills, maintaining appropriate fuel levels may be difficult under hypoxic conditions. The subject must also listen for audio messages addressed to their communications call-sign, which is displayed at the top of the communications window, and ignore messages directed at other call-signs. The audio message directs the subjects to change the frequency of one of the radios listed on the screen, but because hypoxia can negatively impact the subject's ability to learn and memorize their call-sign as well as their ability to pay attention to the audio signal, performance on this task may decrease with the onset of a hypoxic state.

2.2. Physiological and MATB data

The MATB performance variables are updated and reported every 10 s, and oxygen saturation is sampled at 256 Hz. Therefore, our design focused on a window size with a minimum of 10 s and a maximum of 60 s. This multi-resolution windowing approach was designed because of the multi-modal physiological time series data (e.g., ECG, EEG). Specifically, these physiological signals were captured using different windows of time since not all physiological features can provide precise information within such a narrow window (e.g., Heart Rate Complexity Measurements [27]). Thus, not all the time-series data can be placed on a single time-scaled window. Importantly, this multi-resolution approach acquires features that provide both longitudinal trend information on the autonomic system (i.e., longer time windows)



Safety Mechanisms for Protection against Over Pressurization:

A. Regulators have built-in pressure relief valve between first and second stage to prevent catastrophic failure.

B. ROBD2 has built-in overpressure detect mechanism to limits pressure in mask to 0.75 PSIG.

C. ROBD2 has built-in mechanical pressure relief valve prevents pressure in mask from exceeding 1 PSIG.

Fig. 1. Environics ROBD2 system designed for inducing hypoxia, without changing atmospheric pressure.

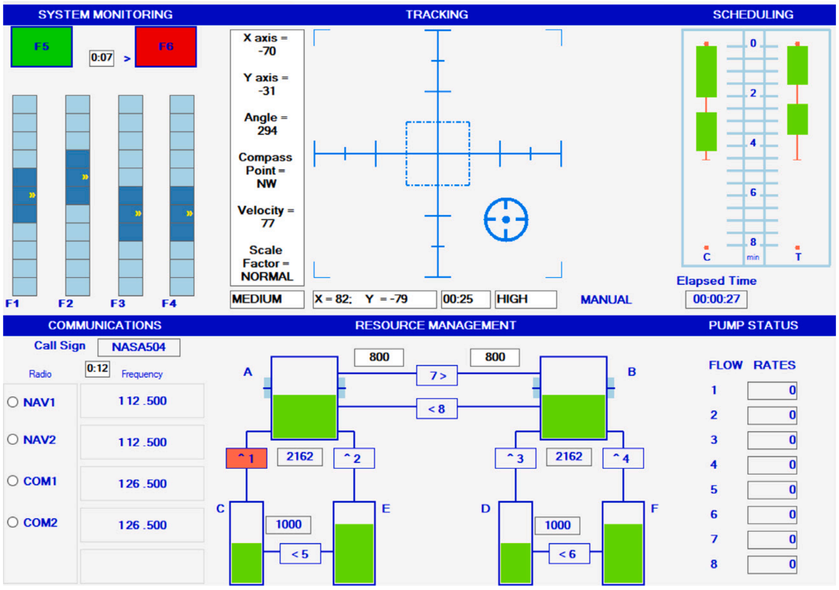


Fig. 2. Illustration of the Multi-Attribute Task Battery (MATB) displayed on the screen of the participant.

and instantaneous features of the autonomic system and cognitive states (i.e., shorter time windows). Intensity analysis was performed, and activation complexity was calculated for EEG waveforms using specialized wavelet filters and complexity measures described in [23].

3. Data structures and augmented ML modeling foundations for fusion

The proposed fusion framework, Naive Adaptive Probabilistic Sensor (NAPS), utilizes numerous small models that have numerous augmented response variables that randomly span the full feature space of our data set, shown in Fig. 4. In this section, we define how to develop the data structure and response variable framework for the proposed fusion framework, which is highlighted in Fig. 4 under the subcomponents of Augmented Responses and ML model.

3.1. Defining the full data set

Utilizing Table 1, a breakdown of the full data in which their modalities and extracted features are denoted as $\mathcal{D}_{\mathcal{S}}$. Matrix $\mathcal{D}_{\mathcal{S}}$ has a size of $(N \times H)$, where the sample of the MATB performance instance is

Table 1
Feature types per sensing modality.

Mod_g	Modality	Feature type	Number of features
Mod_1	ECG	Summary Stats HR complexity	2 2
Mod_2	Respiration	Rate Complexity Interactions	1 2 7
Mod_3	0^2 Saturation	Mean Interactions	1 7
Mod_4	Demographics	Anatomical Flight Info.	2 2
$\text{Mod}_{(5-20)}$	EEG	Power spectrum Engagement index PE complexity	15 × 16 CHL 1 × 16 CHL 1 × 16 CHL
Total			**298**

$\forall\ n \in \{1, 2, \dots, N\}$ and the physiological features (e.g., heart rate (HR) complexity) are $\forall\ h \in \{1, 2, \dots, H\}$. The structure of $\mathcal{D}_\mathbb{S}$ is composed of $G = 20$ modalities (e.g., ECG, respiration, etc.), where each EEG channels is considered its own modality. We will denote these specific modalities as Mod_g, where $\forall\ g \in \{1, 2, \dots, G\}$. The entire data set is defined around Table 1, taking the form

$$\mathcal{D}_\mathbb{S} = [\ \overbrace{\text{Mod}_\mathbf{1}}^{L_1=4}\quad \overbrace{\text{Mod}_\mathbf{2}}^{L_2=10}\quad \dots\quad \overbrace{\text{Mod}_\mathbf{20}}^{L_{20}=17}\quad \mathbf{R}\ ]$$

$$\mathcal{D}_\mathbb{S} = \begin{bmatrix} f_{1(1-4)} & f_{1(5-14)} & \cdots & f_{1(244-H)} & R_1 \\ f_{2(1-4)} & f_{2(5-14)} & \vdots & f_{2(244-H)} & R_2 \\ \vdots & \ddots & \ddots & \dots & \vdots \\ f_{N(1-4)} & f_{N(5-14)} & \cdots & f_{N(244-H)} & R_N \end{bmatrix}, \tag{1}$$

where R_n is the response variable for the nth sample, f_{nh} is the features that are associated to a specific Mod_g, and L_g is the total number of features which are contained within the Mod_g.

3.2. Novel predictive variable data structures requirements

The entire data set, $\mathcal{D}_\mathbb{S}$, is strategically sampled for the development of various small independent models that have overlapping sets of features from $\mathcal{D}_\mathbb{S}$. We will refer to each one of these models as a “sensor model”, S_m, which is linked to an organized sub-set of the feature spaces, D_m. We define this data matrix as,

$$D_m = \begin{bmatrix} f_{11} & f_{12} & \cdots & f_{1K} \\ f_{21} & f_{22} & \vdots & f_{2K} \\ \vdots & \vdots & \ddots & \vdots \\ f_{N1} & f_{N2} & \cdots & f_{NK} \end{bmatrix}, \tag{2}$$

where f_{nk} is a feature, $\forall\ n \in \{1, 2, \dots, N\}$ describing the number of samples, $\forall\ k \in \{1, 2, \dots, K\}$ is the number of features with S_m, and $K \ll H$. A plethora of D_m’s are created for a total of M sensor models in which the data structure, D_m, is restructured as a deep copy data structure,

$$\mathcal{D}_C = [D_1, D_2, \dots, D_{M-1}, D_M], \tag{3}$$

where $\mathcal{D}_C$ contains M data structures associated with their respective M sensor models.

3.2.1. Structuring of a sensor model

The structuring of S_m depends on the number of samples within the data set, the number of response variables (e.g., binary vs. multi-class), and the total number of features within $\mathcal{D}_\mathbb{S}$. These factors all play a role in determining the number of features, K, utilized in creating D_m. The K features used within a “sensor model” aims to maintain an adequate ratio of observations to estimated parameters. Over the years, there have been numerous “rules of thumb” which discuss minimum ratios necessary for binomial distributions, regressions, etc., which have ranged from ratios of 20:1 (observations to one estimated parameter) to ratios of 5:1 [28–31]. Once a K is determined, we do not completely select any random vectors within $\mathcal{D}_\mathbb{S}$’s feature space as a predictive variable. We design a quasi-random structure based on the specific modalities, Mod_g, previously defined. Every “sensor model” from a physiological point of view requires multiple diverse physiological modalities to triangulate a human’s state, and each modality has numerous inputs. Therefore, we want to guarantee that each model will always incorporate a diverse set of specific modalities with structured relationships consisting of randomized features (e.g., eight random EEG features from eight different channels).

3.2.2. Our study sensor model paradigm

Within this study, M sensor models (i.e., feature set), $\{S_m | m = 1, \dots, M\}$, were created from the full vector space, where 11 features were randomly chosen to be placed within the model’s corresponding quasi-random structure D_m. These 11 features for each sensor model had a quasi-random structure by choosing two random EEG channels (e.g., P2, F1), where within each EEG channel, we produced 17 features. However, only 4 EEG features are randomly taken from each of the two random EEG channels. The first part of the quasi-random structure produces a total of 8 EEG random features that are specific to the two random EEG channels that were chosen. The second part of the quasi-random structure chooses three additional random features where one random feature comes from the ECG modality (4 possible features), and two random features come from Respiration, O2 Saturation, or Demographics modalities (22 possible features). This quasi-random structure of feature inputs produces a total of 11 features for each S_m. From the set of D_m’s, we produced a total of $M = 175$ sensor models S_m. Thus, there is obvious overlap within the feature space between various sensors. Reducing the number of features per model will now allow us to avoid model sparsity problems (issues of uncertainty) caused by the ratio of the number of features to the number of observations.

Utilizing the MATB, the response variables are formed based on the performance of their tracking and communications tasks. This developed a 4-class classification dataset for the for proposed algorithms training and testing. These response variables are considered a singleton proposition (discussed later in the Dempster–Shafer framework section). These propositions are no-error (θ_1, $N = 663$ samples), delay in communication task (θ_2, $N = 147$ samples), tracking deviation error (θ_3, $N = 144$ samples), and radio error (θ_4, $N = 45$ samples).

Our study data produced a total of 958 observations of when the pilots were supposed to engage to perform a functional task. The model dimension of 11 features (analogous to the number of parameters we need to estimate) utilizing the 958 observations produces an approximate ratio of 87:1. We extend the ratio rule of thumb because our response variable is not binary, and we are handling a class balance problem, in which down-sampling can potentially occur during the modeling process.

3.3. Augmented response variable combinations and mapping

The framework leverages DS theory’s foundation of examining the full set of mutually exclusive and exhaustive propositions (e.g., class labels) of interest, which is referred to as the *frame of discernment (FOD)*. Thus, for our specific case, the FOD Ω is defined with the four propositions,

$$\Omega = \{\theta_1, \theta_2, \theta_3, \theta_4\}, \tag{4}$$

based on our four class ML Detection problem. Thus, the FOD is a set of all subsets of Ω, which creates the power set 2^Ω. Therefore, the power set consists of the combinatorial sets or propositions that make up Ω. These combinatorial sets allow us to model ignorance related to the ML class label, where the ML model can be trained around various augmented response variables consisting of

$$\begin{aligned} 2^{|\Omega|} \longmapsto \{&\emptyset,\ \{\theta_1\}, \{\theta_2\}, \{\theta_3\}, \{\theta_4\}, \{\theta_1, \theta_2\}, \\ &\{\theta_3, \theta_4\}, \{\theta_2, \theta_3\}, \{\theta_1, \theta_4\}, \{\theta_1, \theta_3\} \\ &\{\theta_2, \theta_4\}, \{\theta_1, \theta_2, \theta_3\}, \{\theta_2, \theta_3, \theta_4\}, \\ &\{\theta_1, \theta_3, \theta_4\}, \{\theta_1, \theta_2, \theta_4\}, \Omega\} \end{aligned} \tag{5}$$

combinations of the response variable. The power set in Eq. (5) demonstrates there are 16 different propositions that can be created. These propositions can be augmented and mixed to produce different mappings of ML models from its associated response variables. Based on the power set in Eq. (5), Table 2 formulates these DST propositions into various augmented response variables, where each combination

Table 2
Augmented response variables using four classes.

Combination variations	Class/Proposition assignments		Number of classes
X_1	$\{\theta_1\}$ $\{\theta_2\}$	$\{\theta_3\}$ $\{\theta_4\}$	4
X_2	$\{\theta_1\}$	$\{\theta_2,\theta_3,\theta_4\}$	2
X_3	$\{\theta_2\}$	$\{\theta_1,\theta_3,\theta_4\}$	2
X_4	$\{\theta_3\}$	$\{\theta_1,\theta_2,\theta_4\}$	2
X_5	$\{\theta_4\}$	$\{\theta_2,\theta_3,\theta_4\}$	2
X_6	$\{\theta_1,\theta_2\}$	$\{\theta_3,\theta_4\}$	2
X_7	$\{\theta_1,\theta_3\}$	$\{\theta_2,\theta_4\}$	2
X_8	$\{\theta_1,\theta_4\}$	$\{\theta_2,\theta_3\}$	2
X_9	$\{\theta_1,\theta_2\}$	$\{\theta_3\}$ $\{\theta_4\}$	3
X_{10}	$\{\theta_1,\theta_3\}$	$\{\theta_2\}$ $\{\theta_4\}$	3
X_{11}	$\{\theta_1,\theta_4\}$	$\{\theta_2\}$ $\{\theta_3\}$	3
X_{12}	$\{\theta_2,\theta_3\}$	$\{\theta_1\}$ $\{\theta_4\}$	3
X_{13}	$\{\theta_2,\theta_4\}$	$\{\theta_1\}$ $\{\theta_3\}$	3
X_{14}	$\{\theta_3,\theta_4\}$	$\{\theta_1\}$ $\{\theta_2\}$	3

variation with the table X_p can be a different mapping of the original recorded response variable, where $\forall\ p \in \{1,2,\dots,P\}$. Therefore, let us consider a single D_m containing N samples of data. A single sample of data, n, can be expressed as

$$D_m(n) = [f_{n1}, f_{n2}, \dots, f_{nK}] \xmapsto{S_m} R_n, \tag{6}$$

where R_n is the recorded and potentially uncertain response variable. The initial R_n's recorded response variable is strictly composed of singleton propositions where the $|\Omega|$ equals C labeled classes, yielding $\{\theta_1\}, \{\theta_2\}, \dots, \{\theta_C\}$ singletons. This produces 2^C-2 potential augmented response variables, where $2^C = P$ for the full set of potential combinations. Given a four class problem, we provide a breakdown of these potential combinations in Table 2, where Ω and the $\emptyset$ elements are excluded within the power set. Each X_p consists of a $\{\cdot\}^+$ proposition that is considered the positive class label for the model, where

$$\{\cdot\}^+ = R_n \cap X_p. \tag{7}$$

Therefore, we can map a sampled response variable R_n as the positive class label for a defined sample of $R_n = \{\cdot\}^+$, which is mapped to a new response variable through the following condition

$$\left\langle S_m(D_m(n)) \xmapsto{R_n} X_p \overset{\text{def}}{=} \left\{ \{\cdot\}^+, \{\mathcal{P}\}^c \right\} \right\rangle \overset{\text{def}}{=} D_{pm}(n), \tag{8}$$

where $\{\mathcal{P}\}^c$ is a proposition or a set of propositions that is the compliment of $\{\cdot\}$ within the FOD, and D_m is extended to a second dimension defining the same predictive variables but are mapped to a different response variable D_{pm}. For an example, let us assume the sampled response variable $R_n = \{\theta_1\}^+$ which will be mapped to the doubleton cases of X_6, X_7, and X_8, shown in Table 2. This mapping to new response variables will produce the following:

$$\begin{aligned} D_{6m}(n) &\overset{\text{def}}{=} \left\langle S_m(D_m(n)) \xmapsto{R_n} X_6 \right\rangle, \\ &\quad \text{where } X_6 \overset{\text{def}}{=} \left\{ \{\theta_1,\theta_2\}^+, \{\theta_3,\theta_4\} \right\} \\ D_{7m}(n) &\overset{\text{def}}{=} \left\langle S_m(D_m(n)) \xmapsto{R_n} X_7 \right\rangle, \\ &\quad \text{where } X_7 \overset{\text{def}}{=} \left\{ \{\theta_1,\theta_3\}^+, \{\theta_2,\theta_4\} \right\} \\ D_{8m}(n) &\overset{\text{def}}{=} \left\langle S_m(D_m(n)) \xmapsto{R_n} X_8 \right\rangle, \\ &\quad \text{where } X_8 \overset{\text{def}}{=} \left\{ \{\theta_1,\theta_4\}^+, \{\theta_2,\theta_3\} \right\}. \end{aligned} \tag{9}$$

These new data sets, D_{pm}, create new simplistic classification boundaries to be relaxed or simply neglected when the response variables are merged together. These new response variables are inputs into the sensor model, S_m. However, these augmented response variables and their respective predictive variables create a new ML Model, $\mathcal{M}_{pm}$. This leads to a new BoE for the fusion framework to account for based on $\mathcal{M}_{pm}$ individual models. The complete set of models are created by M feature sets (or sensor models, where $M = 175$) and P combinations of the response variables (or augment responses, where P), mirroring the new data set D_{pm}. This produces $P \times M$ individual models. The variety of these models can therefore be expressed as a matrix of randomly structured models defined as

$$\begin{bmatrix} \mathcal{M}_{11} & \mathcal{M}_{12} & \dots & \mathcal{M}_{1M} \\ \mathcal{M}_{21} & \mathcal{M}_{pm} & \dots & \mathcal{M}_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ \mathcal{M}_{P1} & \mathcal{M}_{P2} & \dots & \mathcal{M}_{PM} \end{bmatrix}. \tag{10}$$

3.3.1. A numerical data structure example

Let us assume a simple wide data set, $\mathcal{D}_{\mathcal{S}}$, that is 12 by 16, where the number of observed samples are $N = 12$ and the number of features is $H = 16$. The data set structure consists of three modalities ($G = 3$), where Mod_1 has 8 feature ($L_1 = 8$), Mod_2 has 4 features ($L_2 = 4$) and Mod_3 has 4 features ($L_3 = 4$). The response variables have weakly supported ground truth that are assigned one of the four class assignments $\{\theta_1\}$, $\{\theta_2\}$, $\{\theta_3\}$, and $\{\theta_4\}$. We define this data structure as,

$$\begin{aligned} \mathcal{D}_{\mathcal{S}} &= \left[\overbrace{\mathbf{Mod_1}}^{L_1=8} \quad \overbrace{\mathbf{Mod_2}}^{L_2=4} \quad \overbrace{\mathbf{Mod_3}}^{L_3=4} \quad \mathbf{R} \right] \\ &= \left[\ \boldsymbol{f}_{(1-8)} \quad \boldsymbol{f}_{(9-12)} \quad \boldsymbol{f}_{(13-16)} \quad \mathbf{R} \ \right] \end{aligned}$$

$$= \begin{array}{l} \begin{array}{ccccccccc} \boldsymbol{f}_1 & \boldsymbol{f}_2 & \boldsymbol{f}_3 & \boldsymbol{f}_4 & \boldsymbol{f}_5 & \boldsymbol{f}_6 & \boldsymbol{f}_7 & \boldsymbol{f}_8 & \\ 0.37 & 0.18 & 0.61 & 0.66 & 0.39 & 0.73 & 0.12 & 0.25 & \dots \\ 0.95 & 0.30 & 0.17 & 0.31 & 0.27 & 0.77 & 0.71 & 0.41 & \dots \\ 0.73 & 0.52 & 0.07 & 0.52 & 0.83 & 0.07 & 0.76 & 0.76 & \dots \\ 0.60 & 0.43 & 0.95 & 0.55 & 0.36 & 0.36 & 0.56 & 0.23 & \dots \\ 0.16 & 0.29 & 0.97 & 0.18 & 0.28 & 0.12 & 0.77 & 0.08 & \dots \\ 0.16 & 0.61 & 0.81 & 0.97 & 0.54 & 0.86 & 0.49 & 0.29 & \dots \\ 0.06 & 0.14 & 0.30 & 0.78 & 0.14 & 0.62 & 0.52 & 0.16 & \dots \\ 0.87 & 0.29 & 0.10 & 0.94 & 0.80 & 0.33 & 0.43 & 0.93 & \dots \\ 0.60 & 0.37 & 0.68 & 0.89 & 0.07 & 0.06 & 0.03 & 0.81 & \dots \\ 0.71 & 0.46 & 0.44 & 0.60 & 0.99 & 0.31 & 0.11 & 0.63 & \dots \\ 0.02 & 0.79 & 0.12 & 0.92 & 0.77 & 0.33 & 0.03 & 0.87 & \dots \\ 0.97 & 0.20 & 0.50 & 0.09 & 0.20 & 0.73 & 0.64 & 0.80 & \dots \end{array} \\ \\ \begin{array}{cccccccccc} & \boldsymbol{f}_9 & \boldsymbol{f}_{10} & \boldsymbol{f}_{11} & \boldsymbol{f}_{12} & \boldsymbol{f}_{13} & \boldsymbol{f}_{14} & \boldsymbol{f}_{15} & \boldsymbol{f}_{16} & \mathbf{R} \\ \dots & 0.81 & 0.32 & 0.91 & 0.84 & 0.34 & 0.35 & 0.55 & 0.24 & \theta_3 \\ \dots & 0.90 & 0.52 & 0.24 & 0.32 & 0.11 & 0.73 & 0.69 & 0.97 & \theta_4 \\ \dots & 0.32 & 0.70 & 0.14 & 0.19 & 0.92 & 0.90 & 0.65 & 0.39 & \theta_3 \\ \dots & 0.11 & 0.36 & 0.49 & 0.04 & 0.88 & 0.89 & 0.22 & 0.89 & \theta_2 \\ \dots & 0.23 & 0.97 & 0.99 & 0.59 & 0.26 & 0.78 & 0.71 & 0.63 & \theta_1 \\ \dots & 0.43 & 0.96 & 0.24 & 0.68 & 0.66 & 0.64 & 0.24 & 0.79 & \theta_3 \\ \dots & 0.82 & 0.25 & 0.67 & 0.02 & 0.82 & 0.08 & 0.33 & 0.50 & \theta_4 \\ \dots & 0.86 & 0.50 & 0.76 & 0.51 & 0.56 & 0.16 & 0.75 & 0.58 & \theta_1 \\ \dots & 0.01 & 0.30 & 0.24 & 0.23 & 0.53 & 0.90 & 0.65 & 0.49 & \theta_3 \\ \dots & 0.51 & 0.28 & 0.73 & 0.65 & 0.24 & 0.61 & 0.85 & 0.20 & \theta_4 \\ \dots & 0.42 & 0.04 & 0.37 & 0.17 & 0.09 & 0.01 & 0.66 & 0.72 & \theta_2 \\ \dots & 0.22 & 0.61 & 0.63 & 0.69 & 0.90 & 0.10 & 0.57 & 0.28 & \theta_3 \end{array} \end{array}.$$

For the above $\mathcal{D}_{\mathcal{S}}$, $M = 50$ sensors models are created using D_m, an organized subset of features from $\mathcal{D}_{\mathcal{S}}$. This is reorganized as a deep copy data structure,

$$\mathcal{D}_C = [D_1, D_2, \dots, D_{M-1}, D_{50}], \tag{11}$$

where $\mathcal{D}_C$ contains 50 data structures associated with their respective 50 sensor models. This allows us to account for high dimensionality by using small models that span the features space. For the simplicity of this example, we will only discuss D_1 and D_2 and use a reduced observations-to-features ratio of 3:1. However, note that in practical applications this ratio should at minimum be in the range of 5:1 through 20:1. Based the quasi-random sampling scheme for D_m, as discussed in Section 3.2.1, we sample 2 features from Mod_1, 1 feature from Mod_2, and 1 feature from Mod_3. Thus, we form

$$D_1 = \begin{bmatrix} \overbrace{0.39}^{\boldsymbol{f}_5} & \overbrace{0.73}^{\boldsymbol{f}_6} & \overbrace{0.84}^{\boldsymbol{f}_{12}} & \overbrace{0.35}^{\boldsymbol{f}_{14}} \\ 0.27 & 0.77 & 0.32 & 0.73 \\ 0.83 & 0.07 & 0.19 & 0.90 \\ 0.36 & 0.36 & 0.04 & 0.89 \\ 0.28 & 0.12 & 0.59 & 0.78 \\ 0.54 & 0.86 & 0.68 & 0.64 \\ 0.14 & 0.62 & 0.02 & 0.08 \\ 0.80 & 0.33 & 0.51 & 0.16 \\ 0.07 & 0.06 & 0.23 & 0.90 \\ 0.99 & 0.31 & 0.65 & 0.61 \\ 0.77 & 0.33 & 0.17 & 0.01 \\ 0.20 & 0.73 & 0.69 & 0.10 \end{bmatrix} \quad D_2 = \begin{bmatrix} \overbrace{0.61}^{\boldsymbol{f}_3} & \overbrace{0.39}^{\boldsymbol{f}_5} & \overbrace{0.81}^{\boldsymbol{f}_9} & \overbrace{0.35}^{\boldsymbol{f}_{14}} \\ 0.17 & 0.27 & 0.90 & 0.73 \\ 0.07 & 0.83 & 0.32 & 0.90 \\ 0.95 & 0.36 & 0.11 & 0.89 \\ 0.97 & 0.28 & 0.23 & 0.78 \\ 0.81 & 0.54 & 0.43 & 0.64 \\ 0.30 & 0.14 & 0.82 & 0.08 \\ 0.10 & 0.80 & 0.86 & 0.16 \\ 0.68 & 0.07 & 0.01 & 0.90 \\ 0.44 & 0.99 & 0.51 & 0.61 \\ 0.12 & 0.77 & 0.42 & 0.01 \\ 0.50 & 0.20 & 0.22 & 0.10 \end{bmatrix}. \tag{12}$$

Using D_1 as an exemplar, we construct the augmentation of the response variables using the combinations X_8 and X_9. In practical applications, we suggest to expand to a larger set of combinations and examine the performance of NAPS Fusion with different balanced variations. Utilizing these augmented response combinations, D_1 will help generate two augmented response data sets D_{pm} via Eq. (8). These D_{pm} are

$$D_{81} = \begin{bmatrix} \overbrace{0.39}^{\boldsymbol{f}_5} & \overbrace{0.73}^{\boldsymbol{f}_6} & \overbrace{0.84}^{\boldsymbol{f}_{12}} & \overbrace{0.35}^{\boldsymbol{f}_{14}} & \overbrace{\{\theta_2,\theta_3\}^+}^{\{\cdot\}^+} \\ 0.27 & 0.77 & 0.32 & 0.73 & \{\theta_1,\theta_4\}^+ \\ 0.83 & 0.07 & 0.19 & 0.90 & \{\theta_2,\theta_3\}^+ \\ 0.36 & 0.36 & 0.04 & 0.89 & \{\theta_2,\theta_3\}^+ \\ 0.28 & 0.12 & 0.59 & 0.78 & \{\theta_1,\theta_4\}^+ \\ 0.54 & 0.86 & 0.68 & 0.64 & \{\theta_2,\theta_3\}^+ \\ 0.14 & 0.62 & 0.02 & 0.08 & \{\theta_1,\theta_4\}^+ \\ 0.80 & 0.33 & 0.51 & 0.16 & \{\theta_1,\theta_4\}^+ \\ 0.07 & 0.06 & 0.23 & 0.90 & \{\theta_2,\theta_3\}^+ \\ 0.99 & 0.31 & 0.65 & 0.61 & \{\theta_1,\theta_4\}^+ \\ 0.77 & 0.33 & 0.17 & 0.01 & \{\theta_2,\theta_3\}^+ \\ 0.20 & 0.73 & 0.69 & 0.10 & \{\theta_2,\theta_3\}^+ \end{bmatrix} \quad D_{91} = \begin{bmatrix} \overbrace{0.39}^{\boldsymbol{f}_5} & \overbrace{0.73}^{\boldsymbol{f}_6} & \overbrace{0.84}^{\boldsymbol{f}_{12}} & \overbrace{0.35}^{\boldsymbol{f}_{14}} & \overbrace{\{\theta_3\}^+}^{\{\cdot\}^+} \\ 0.27 & 0.77 & 0.32 & 0.73 & \{\theta_4\}^+ \\ 0.83 & 0.07 & 0.19 & 0.90 & \{\theta_3\}^+ \\ 0.36 & 0.36 & 0.04 & 0.89 & \{\theta_1,\theta_2\}^+ \\ 0.28 & 0.12 & 0.59 & 0.78 & \{\theta_1,\theta_2\}^+ \\ 0.54 & 0.86 & 0.68 & 0.64 & \{\theta_3\}^+ \\ 0.14 & 0.62 & 0.02 & 0.08 & \{\theta_4\}^+ \\ 0.80 & 0.33 & 0.51 & 0.16 & \{\theta_1,\theta_2\}^+ \\ 0.07 & 0.06 & 0.23 & 0.90 & \{\theta_3\}^+ \\ 0.99 & 0.31 & 0.65 & 0.61 & \{\theta_4\}^+ \\ 0.77 & 0.33 & 0.17 & 0.01 & \{\theta_1,\theta_2\}^+ \\ 0.20 & 0.73 & 0.69 & 0.10 & \{\theta_3\}^+ \end{bmatrix}, \tag{13}$$

where the augmented response variables are provided in the last column. The original responses are mapped to augmented responses using Eq. (8), where for

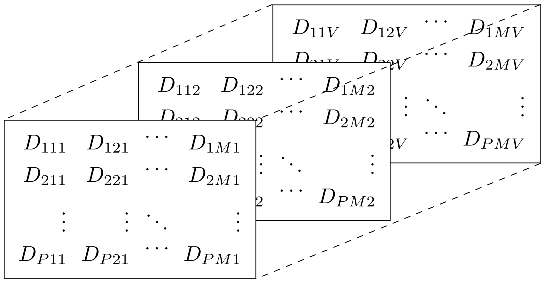
$$\begin{aligned} D_{81}(1) :&\quad R_1 = \theta_3 \mapsto \{\theta_2,\theta_3\}^+, \\ D_{81}(2) :&\quad R_2 = \theta_4 \mapsto \{\theta_1,\theta_4\}^+, \\ D_{91}(1) :&\quad R_1 = \theta_3 \mapsto \{\theta_3\}^+, \\ D_{91}(2) :&\quad R_2 = \theta_4 \mapsto \{\theta_4\}^+. \end{aligned}$$

These newly formed D_{pm} data sets will be used to train and test the models $\mathcal{M}_{81}$ and $\mathcal{M}_{91}$. We can note how each nth sample incorporates its respective R_n response variable within the augmented response combination. This allows the models to relax the boundary constraints for the various augmented response combinations.

3.4. ML model development for body of evidence (BoE)

For such an evidence fusion approach, we are constrained to specific types of ML models that may not generalize well to probabilistic frameworks. We avoid this issue by utilizing a bagging approach, also known as bootstrap aggregation. A bagging approach not only improves the stability and accuracy of the ML model (reducing high variance/uncertainty) for the statistical classification but can be easily thought of as a probability structure. The typical bagging approach subsamples the observations of $\mathcal{M}_{pm}$'s vector space, T times, using sampling with replacement (typically around 60% of the data). Therefore, for each $\mathcal{M}_{pm}$, we essentially create T "micro-models", since $\mathcal{M}_{pm}$ is already a subspace of the original data set. The T models are trained and produce a frequency for each time a class was chosen as the predicted class and could be thought of as a simplistic probability structure.

Therefore, each BoE can eventually be formed into probabilistic "supports" that are obtained from the outputs of each ML model's, $\mathcal{M}_{pm}$, that are associated to Table 2. Therefore, for each $\mathcal{M}_{pm}$ the output is utilized to provide "supports" that are linked to $\{\cdot\}^+$ and $\{\mathcal{P}\}^c$ elements within the power set defined by,



D_{11V} D_{12V} $\cdots$ D_{1MV}
D_{2MV}
D_{PMV}
D_{112} D_{122} $\cdots$ D_{1M2}
D_{2M2}
D_{PM2}
D_{111} D_{121} $\cdots$ D_{1M1}
D_{211} D_{221} $\cdots$ D_{2M1}
D_{P11} D_{P21} $\cdots$ D_{PM1}

Fig. 3. A data structure representation reconstructed data set, D_{pmv}, for sensor models (M), response combinations (P), and Cross-Validation (V).

$$\mathcal{V}_{pm}(n) = \begin{bmatrix} V_1 & V_2 & \cdots & V_{(P-2)} \end{bmatrix}^T, \tag{14}$$

where V_i is the number of votes from the bagged ML model from T total votes, $\{\cdot\}^+$ and $\{\mathcal{P}\}^c$ are where the "supports" are given for the elements linked to the specific V_i elements within $V_{pm}(n)$ for sample n within the data set. When $V_i = 0$ that element within $\mathcal{V}_{pm}(n)$ will not be a focal element for that specific data set instance, $D_{pm}(n)$,

3.5. Expanding the data structure for Cross-Validation

For our implemented framework, we only explored binary augmented response variable models and the classical 4-class problem (combinations 1–8 in Table 2), where the total number of combinations is $P = 8$. Cases 1–8 were utilized to explore the extreme cases of modeling ignorance versus the classical approach. This expands our definition of Eq. (3), to be two-dimensional where the deep copy data structure element D_m is expanded to D_{pm} where p is a combination of the response variable and P is the total number of combinations. Augmenting the response is critical to understanding its impact on the number of observations within your newly formed classes. Depending on the combinations that are chosen as your response variables $(\{\{\theta_1,\theta_3,\theta_4\},\{\theta_2\}\})$, the class imbalances within the ML paradigm will improve or can severely worsen. In order to attenuate this issue of the class imbalances causing model bias, Θ_2, Synthetic Minority Oversampling Technique (SMOTE) is dynamically implemented for each individualized matrix, D_{pm} [32–34]. If a cross-validation (C.V.) approach is used in which the data is partitioned (C.V. was applied to this study), the deep copy data structure is expanded to three dimensions, where D_{pm} is expanded to D_{pmv} (see in Fig. 3).

4. Naive Adaptive Probabilistic Sensor (NAPS) fusion methodology

4.1. Generalizing ML frameworks for DST

The proposed framework can be applied to any ML approach that outputs probabilities, distances, or a voting paradigm in which conflict and "support" can be measured between different propositions within a single form of evidence (i.e., an ML model). The NAPS framework stacks the DS framework over many ML models, which requires the outputs of the ML model to fit within a DS Framework, where a *basic probability assignment (BPA),* otherwise referred to as a *mass assignment,* is created in which contextual considerations (e.g., source reliability, source conflicts, etc.) all play a role in determining the mass to be allocated to a given proposition A_i [18]. The mass assignment is a function of $m : 2^{\Omega} \to [0,1]$, where 2^{Ω} is the power set, such that

$$m(\emptyset) = 0; \qquad \sum_{A_i \subseteq 2^{\Omega}} m(A_i) = 1. \tag{15}$$

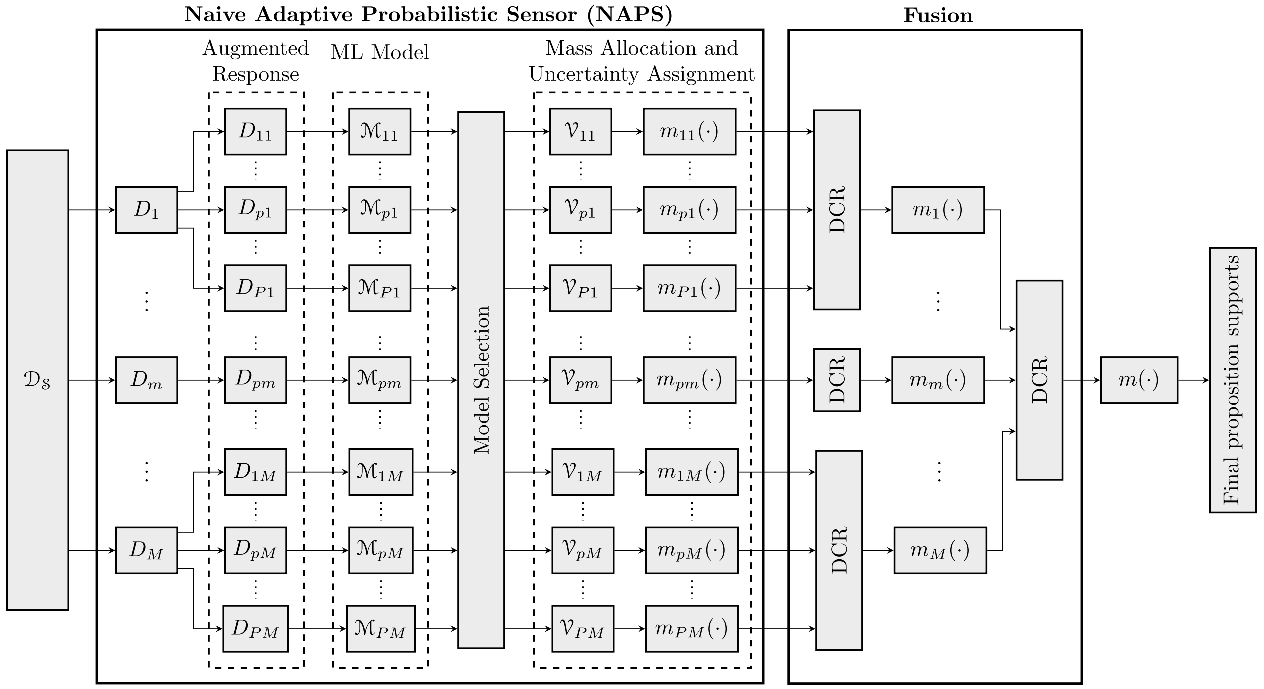


Fig. 4. This block diagram outlines the major steps within NAPS fusion that enable the modeling approach to overcome dimensionality hurdles, potential ambiguous class labels during training, improve modality for new sensors, and adapt to each sample independently, allowing for improved subject-to-subject variability.

Any proposition that is allocated a non-zero mass is referred to as a *focal element*. The *core* $\mathfrak{F}$ refers to the set of focal elements and the *BoE* $\mathcal{E}$ refers to the triplet $\{\Theta, \mathfrak{F}, m(\cdot)\}$. Each A_i within $\mathcal{E}$ is linked to a combination of the response variable produced by the output of $\mathcal{M}_{pm}$, which P combinations are created by augmenting the responses. These augmented response variables and their respective predictive variables are placed into an ML paradigm, shown in Fig. 3, leading to its own BoE created by $\mathcal{M}_{pm}$ individual models. The complete set of models are created by M feature sets (or sensor models, where $M = 175$) and P combinations of the response variables (or augment responses). This produces $M \times P$ individual models, $\mathcal{M}_{PM}$, defined in Eq. (10). *Dempster's combination rule (DCR)* allows one to combine or fuse evidence represented as DST models [35]:

$$m(A) = \frac{\sum_{B\cap C=A} m_1(B)\, m_2(C)}{1 - \sum_{B\cap C=\emptyset} m_1(B)\, m_2(C)}, \tag{16}$$

where $\mathcal{E}_1 = \{\Omega, \mathfrak{F}_1, m_1(\cdot)\}$ and $\mathcal{E}_2 = \{\Omega, \mathfrak{F}_2, m_2(\cdot)\}$ are the BoEs being fused to generate the fused BoE $\mathcal{E} = \{\Omega, \mathfrak{F}, m(\cdot)\}$. The fused mass and BoE generated by the DCR are usually denoted by $m = m_1 \oplus m_2$ and $\mathcal{E} = \mathcal{E}_1 \oplus \mathcal{E}_2$, respectively. The DCR is commutative and associative, thus allowing one to fuse multiple sources of evidence with ease. In order to apply DCR, we have properly designed a mass function to quantify the uncertainty and support of the propositions.

4.2. Mass allocation and uncertainty assignment

In order to fully develop a BPA from a BoE (i.e. $\mathcal{M}_{pm}$), an uncertainty assignment is required to utilize the DST framework approach. The uncertainty assignment is obtained by accounting for two factors: (1) The uncertainty on the contextual meaning of the decision; (2) The uncertainty of the model biasing for the augmented classes. Once the uncertainty is calculated for the specific model, we are able to re-update the “supports” for the model and complete the BPA from the BoE. These BPA are then selected and combined to determine the final supports for a specific instance within the data.

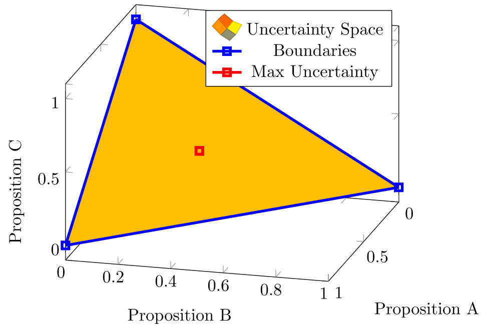


Fig. 5. A visualization of the uncertainty space.

4.2.1. Uncertainty on the contextual meaning of the decision

We obtain contextual meaning based on the frequency of each class. This provides informative context for predicting a specific class based on which T models were reported as the predicted class within the bagging. For example, if we have four classes and 100 models and receive 25 votes for each class, we become very uncertain about the appropriate class to pick. Thus, the model for this specific instance is unreliable. Standard ensemble methods do not contextualize the model within the voting and strictly count the amount of votes [36] and do not look at conflict between the propositions within the model and across models. The model, $\mathcal{M}_{pn}$, when compared to model $\mathcal{M}_{p(m+1)}$ that handles a different set of sensors, may have a different amount of conflict. Fusing these models together can be thought of as a way to easily perform a pseudo method of model averaging, which weights each model differently according to the uncertainty assignment.

Thus, we aim to model this uncertainty through the normalized bagging voting space, where T_{Tot} is the total number of votes (bags in the model) and T_c is the number of tallied votes received for the cth class where C is the total number of classes defined by the x_i combination variation, shown in Table 2. Therefore, we can define a bounded vector space defined by the number of potential classes the

ML model is built around (this is independent on the number of focal elements) and the amount of votes each hypothesis received by

$$\Theta_1 = 1 - \frac{\sqrt{\left(\frac{T_1}{T_{Tot}} - \frac{1}{C}\right)^2 + \cdots + \left(\frac{T_C}{T_{Tot}} - \frac{1}{C}\right)^2}}{\sqrt{\frac{C-1}{C}}}. \tag{17}$$

Eq. (17) is visually depicted in Fig. 5, where, if all the proportions of the votes are approximately equal, the uncertainty assignment operates near the maximum uncertainty region in red. Conversely, if the votes are highly disproportional and there is a dominating proposition, the uncertainty operates near the edges of the vector space where the minimum uncertainty resides.

4.2.2. Uncertainty of the biased model

Model uncertainty is also altered when class imbalances are imposed on the model by the data. When classes are imbalanced, the model will tend to favor the majority class more (the prior). Therefore, similar to the previous uncertainty assignment, this adds additional doubt as to the model's ability to perform. Although SMOTE can be applied to address the class imbalance, the approach is not always able to fully rectify the class imbalance, and the samples are still synthetic instances. Therefore, we use the original data sets for our uncertainty assignment to address the class imbalance issues. The imbalance of the class depends on the proportion of the number of instances that occur for the class, I_c, over the total instances in the data set, I_{Tot}. Thus, quantifying the uncertainty associated to an imbalance for a multi-class problem is defined as,

$$\Theta_2 = \frac{\sqrt{\left(\frac{I_1}{I_{Tot}} - \frac{1}{C}\right)^2 + \cdots + \left(\frac{I_C}{I_{Tot}} - \frac{1}{C}\right)^2}}{\sqrt{\frac{C-1}{C}}}. \tag{18}$$

4.2.3. Accounting for total uncertainty

The total uncertainty, Θ, for the BoE is calculated by simply taking the mean of the two methods that account for the uncertainty in the BoE through,

$$m(\Omega) = \frac{1 - e^{\frac{-\frac{1}{2}(\Theta_1 + \Theta_2)}{d}}}{1 - e^{-1/d}} \tag{19}$$

where d determines the exponential weighting of the distance in the uncertainty space by

$$d = (e^{1-\rho} - 1)^{e(1)}, \tag{20}$$

and ρ is the hyperparameter that determines the strength of the exponential weighing. The hyperparameter ρ is bounded [0,1]. When $\rho = 0$ the function is nearly linear and as ρ approaches 1 the function exponentially grows. We allocate the mass across the full set of propositions $m(A_i)$ over T_{Tot} votes and arrive at the following DST model:

$$m(A) = \begin{cases} \frac{\mathcal{V}_{pm}(n)}{T_{Tot}} \cdot (1 - m(\Omega)), & \text{for } A = V_i; \\[2ex] \frac{1 - e^{\frac{-\frac{1}{2}(\Theta_1 + \Theta_2)}{d}}}{1 - e^{-1/d}}, & \text{for } A = \Omega; \\[2ex] 0, & \text{otherwise.} \end{cases} \tag{21}$$

4.3. Model selection

Due to the nature of DST's framework and cognitive state experiments, developing a model requires synergy for it to work well with the psychophysiology data and for us to fuse models together. Since the sensors (subsets of features) were generated randomly, not all the sensors will capture the appropriate amount of variance to be a strong predictor. In addition, a top-performing sensor may no longer be the sensor for a different response combination (a different augment class). Similar to how the uncertainty adapts and varies across the samples, the sensors used for each combination of the model should also adapt. In order to accomplish this, we took a simplistic approach where we utilized the six sensors with the lowest uncertainty for each response combination. Therefore, if we implement an all-vs-one method (Response Combinations 2–5), each response combination has six sensors that are combined, producing 24 different models for the decision process. All the selected models are combined to account for a large feature space, thus acting like a larger complex model.

An all-vs-one approach is a simplistic and contemporary approach that uses a binarization strategy, where we build a binary response model for each class [37]. The generated combinatorial responses for $\mathcal{M}_{2m}$ to $\mathcal{M}_{5m}$ are utilized $\{\{\theta_1, \theta_2, \theta_3\}, \{\theta_4\}\}$, $\{\{\theta_1, \theta_2, \theta_4\}, \{\theta_3\}\}$, $\{\{\theta_1, \theta_3, \theta_4\}, \{\theta_2\}\}$, or $\{\{\theta_2, \theta_3, \theta_4\}, \{\theta_1\}\}$. All the models are evaluated and the one with the highest probability is the predicted class [38]. This addresses the problem with ignorance but the problem of feature importance and dimensionality reduction still exists. For instance, when you have a four-class, classic singleton case we would generate $\{\theta_1\}$, $\{\theta_2\}$, $\{\theta_3\}$, $\{\theta_4\}$. However, for a non-traditional one-vs-all ML framework design, we would generate combinatorial responses such as $\{\{\theta_1, \theta_2, \theta_3\}, \{\theta_4\}\}$, or $\{\{\theta_1, \theta_2, \theta_4\}, \{\theta_3\}\}$, or $\{\{\theta_1, \theta_3, \theta_4\}, \{\theta_2\}\}$, or $\{\{\theta_2, \theta_3, \theta_4\}, \{\theta_1\}\}$. We then evaluated which key features for the model will be analyzed and how models perform over these combinational patterns.

5. A numerical fusion example

Data and Mass Assignment for $\mathcal{E}_1$: Let us consider the first body of evidence, $\mathcal{E}_1$, with the data set D_{9m1} to predict the nth sample of data using sensor model, S_m, where the response variable from the data set is $R_n = \{\theta_3\}$ for a 10-fold cross validation problem and $\rho = 0$. The response variable, R_n, is mapped to the response combination X_9, created by

$$D_{9m}(n) \overset{\text{def}}{=} \left\langle S_m(D_m(n)) \xrightarrow{R_n} X_9 \right\rangle,$$

where $X_9 \overset{\text{def}}{=} \left\{\{\cdot\}^+, \{\mathcal{P}\}^c\right\}$, the positive class $\{\cdot\}^+ \overset{\text{def}}{=} \{\theta_3\}$, and the compliment sets are $\{\mathcal{P}\}^c \overset{\text{def}}{=} \{\{\theta_1, \theta_2\}, \{\theta_4\}\}$. Let us consider a random forest bagging algorithm that produces that $\mathcal{V}_{pm}(n)$ where $P = 14$ and $T = 150$, producing a set of potential propositions,

$$\mathcal{V}_{9m}(n) = \begin{bmatrix} V_1 \\ \vdots \\ V_3 \\ V_4 \\ \vdots \\ V_8 \\ \vdots \\ V_{14} \end{bmatrix} = \begin{bmatrix} \{\theta_1\} \\ \vdots \\ \{\theta_1, \theta_2\} \\ \{\theta_3\} \\ \vdots \\ \{\theta_4\} \\ \vdots \\ \{\theta_2, \theta_3, \theta_4\} \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 100 \\ 25 \\ \vdots \\ 25 \\ \vdots \\ 0 \end{bmatrix}.$$

Thus, we define the focal elements within $\mathcal{E}_1$ as $\{\theta_1, \theta_2\}$, $\{\theta_3\}$, and $\{\theta_4\}$. Based on the class labels of the ML training data set $D_{9m(2-V)}$, our samples for each class of the labels are perfectly balanced where $I_1 = 1000$, $I_2 = 1000$, and $I_3 = 1000$ producing no uncertainty with respect to a class imbalance in the model, $\Theta_2 = 0$. Regarding the uncertainty of the decision boundaries of the model, based on $\mathcal{V}_{9m1}(n)$ we produce $\Theta_1 = .5$. Therefore, utilizing Eq. (21), we define the body of evidence's

Table 3
Reduced mass assignment.

$m_1(\cdot)$	$2^{\lvert\Omega\rvert}$	Proposition	$m_2(\cdot)$
0.4854	V_3	$\{\theta_1,\theta_2\}$	0
0.1214	V_4	$\{\theta_3\}$	0
0	V_6	$\{\theta_2,\theta_3\}$	0.2037
0.1214	V_8	$\{\theta_4\}$	0
0	V_9	$\{\theta_1,\theta_4\}$	0.2037
0.2719	Ω	$\{\theta_1,\theta_2,\theta_3,\theta_4\}$	0.5927

Table 4
Reduced combination.

		$\mathcal{E}_1$			
	$\cap$	V_3	V_4	V_8	Ω
$\mathcal{E}_2$	V_6	$\{\theta_2\}$	$\{\theta_3\}$	$\emptyset$	$\{\theta_2,\theta_3\}$
	V_9	$\{\theta_1\}$	$\emptyset$	$\{\theta_4\}$	$\{\theta_1,\theta_4\}$
	Ω	$\{\theta_1,\theta_2\}$	$\{\theta_3\}$	$\{\theta_4\}$	Ω

uncertainty as $m(\Omega) = .2719$ and our focal elements mass assignment as $m(V_3) = 0.4854$, $m(V_4) = 0.1214$, and $m(V_8) = 0.1214$.

Data and Mass Assignment for $\mathcal{E}_2$: Now let us consider a secondary body of evidence, $\mathcal{E}_2$, with the data set D_{8m1} to predict the nth sample of data using sensor model, S_m, where the response variable from the data set is $R_n = \{\theta_3\}$ for a 10-fold cross validation problem and $\rho = 0$. The response variable, R_n, is mapped to the response combination X_8, created by

$$D_{8m}(n) \overset{\text{def}}{=} \left\langle S_m(D_m(n)) \xrightarrow{R_n} X_8 \right\rangle,$$

where $X_8 \overset{\text{def}}{=} \left\{ \{\cdot\}^+, \{\mathcal{P}\}^c \right\}$, the positive class $\{\cdot\}^+ \overset{\text{def}}{=} \{\theta_2,\theta_3\}$, and the compliment sets are $\{\mathcal{P}\}^c \overset{\text{def}}{=} \{\theta_1,\theta_4\}$. Let us consider a random forest bagging algorithm that produces that $\mathcal{V}_{pm}(n)$ where $P = 14$ and $T = 150$, producing a set of potential propositions,

$$\mathcal{V}_{9m}(n) = \begin{bmatrix} V_1 \\ \vdots \\ V_6 \\ \vdots \\ V_9 \\ \vdots \\ V_{14} \end{bmatrix} = \begin{bmatrix} \{\theta_1\} \\ \vdots \\ \{\theta_2,\theta_3\} \\ \vdots \\ \{\theta_1,\theta_4\} \\ \vdots \\ \{\theta_2,\theta_3,\theta_4\} \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 75 \\ \vdots \\ 75 \\ \vdots \\ 0 \end{bmatrix}.$$

Thus, we define the focal elements within $\mathcal{E}_2$ as $\{\theta_2,\theta_3\}$, and $\{\theta_1,\theta_4\}$. Based on the class labels of the ML training data set $D_{9m(2-V)}$, our samples for each of the class labels are perfectly balanced where $I_1 = 1700$ and $I_2 = 1300$, producing no uncertainty with respects to a class imbalance in the model, $\Theta_2 = .1333$. Regarding the uncertainty of the decision boundaries of the model, based on $\mathcal{V}_{9m1}(n)$ we produce $\Theta_1 = 1$. Therefore, utilizing Eq. (21), we define the body of evidence's uncertainty as $m(\Omega) = 0.5927$ and our focal elements mass assignment as $m(V_6) = 0.2037$, and $m(V_9) = 0.2037$.

Fusion of $\mathcal{E}_1$ and $\mathcal{E}_2$: Denoting the mass assignment of $\mathcal{E}_1$ as m_1 and that of $\mathcal{E}_2$ as m_2, the values obtained from each BoE can be fused by using DCR given in Eq. (16). A summary of the mass assignments is given in Table 3 and the intersections of the propositions for the application of DCR are provided in Table 4.

Utilizing the intersections of these propositions in Table 4, we are able to proceed with the DCR calculation. Thus, within the DCR calculation we will update the support for $\{\theta_1\}$, $\{\theta_2\}$, $\{\theta_3\}$, $\{\theta_4\}$, $\{\theta_1,\theta_2\}$, $\{\theta_2,\theta_3\}$, $\{\theta_1,\theta_4\}$, and Ω, where non-focal elements are omitted. All calculated mass assignments are summarized in Table 5. Let us first start up updating the support for $\{\theta_1\}$, where we have

$$B \cap C = V_3 \cap V_9 = \{\theta_1\} \implies m_1(V_3)m_2(V_9) = 0.0989$$

and

$$\sum_{B\cap C=\{\theta_1\}} m_1(B)\,m_2(C) = 0.0989.$$

For $\{\theta_2\}$, we have

$$B \cap C = V_3 \cap V_6 = \{\theta_2\} \implies m_1(V_3)m_2(V_6) = 0.0989$$

and

$$\sum_{B\cap C=\{\theta_2\}} m_1(B)\,m_2(C) = 0.0989.$$

For $\{\theta_3\}$, we have

$$B \cap C = V_4 \cap V_6 = \{\theta_3\} \implies m_1(V_4)m_2(V_6) = 0.0247$$

$$B \cap C = V_4 \cap \Omega = \{\theta_3\} \implies m_1(V_4)m_2(\Omega) = 0.0720.$$

and

$$\sum_{B\cap C=\{\theta_3\}} m_1(B)\,m_2(C) = 0.0967.$$

For $\{\theta_4\}$, we have

$$B \cap C = V_8 \cap V_9 = \{\theta_4\} \implies m_1(V_8)m_2(V_9) = 0.0247,$$

$$B \cap C = V_8 \cap \Omega = \{\theta_4\} \implies m_1(V_8)m_2(\Omega) = 0.0720.$$

and

$$\sum_{B\cap C=\{\theta_4\}} m_1(B)\,m_2(C) = 0.0967.$$

For $\{\theta_1,\theta_2\}$, we have

$$B \cap C = V_3 \cap \Omega = \{\theta_1,\theta_2\} \implies m_1(V_3)m_2(\Omega) = 0.2877.$$

and

$$\sum_{B\cap C=\{\theta_1,\theta_2\}} m_1(B)\,m_2(C) = 0.2877.$$

For $\{\theta_1,\theta_4\}$, we have

$$B \cap C = \Omega \cap V_9 = \{\theta_1,\theta_4\} \implies m_1(\Omega)m_2(V_9) = 0.0554.$$

and

$$\sum_{B\cap C=\{\theta_1,\theta_4\}} m_1(B)\,m_2(C) = 0.0554.$$

For $\{\theta_2,\theta_3\}$, we have

$$B \cap C = \Omega \cap V_6 = \{\theta_2,\theta_3\} \implies m_1(\Omega)m_2(V_6) = 0.0554.$$

and

$$\sum_{B\cap C=\{\theta_2,\theta_3\}} m_1(B)\,m_2(C) = 0.0554.$$

Finally, for Ω, we have

$$B \cap C = \Omega \cap \Omega = \Omega \implies m_1(\Omega)m_2(\Omega) = 0.1610.$$

and

$$\sum_{B\cap C=\Omega} m_1(B)\,m_2(C) = 0.1610.$$

Table 5
Mass assignment after fusion.

$2^{\|\Omega\|}$	$m(\cdot)$
$\{\theta_1\}$	0.1040
$\{\theta_2\}$	0.1040
$\{\theta_3\}$	0.3027
$\{\theta_4\}$	0.1017
$\{\theta_1, \theta_2\}$	0.0583
$\{\theta_1, \theta_4\}$	0.1017
$\{\theta_2, \theta_3\}$	0.0583
Ω	0.1696

All that remains is to calculate the null set of the DCR calculation which is associated to the denominator

$$1 - \sum_{B \cap C = \emptyset} m_1(B)\, m_2(C). \tag{22}$$

For $\{\emptyset\}$, we have

$$\begin{aligned} B \cap C &= V_8 \cap V_6 = \{\emptyset\} \\ &\implies m_1(V_8)m_2(V_6) = 0.0247, \\ B \cap C &= V_4 \cap V_9 = \{\emptyset\} \\ &\implies m_1(V_4)m_2(V_9) = 0.0247. \end{aligned}$$

Therefore,

$$\sum_{B \cap C = \emptyset} m_1(B)\, m_2(C) = 0.4940,$$

and

$$1 - \sum_{B \cap C = \emptyset} m_1(B)\, m_2(C) = 0.9505.$$

6. Results

In this section, we address the following research questions regarding the use case of implementing Naive Adaptive Probabilistic Sensor Fusion for a more generalized ML framework to handle experimental study data for machine computer interaction:

RQ1 How do typical dimensionality reduction methods perform on our multi-class problems with small samples, class imbalances, and large features?

RQ2 How does dimensionality reduction combined with augmented response perform?

RQ3 As we address these issues caused by uncertainty, how does performance for these ML paradigms change utilizing a NAPS framework?

6.1. RQ1: Dimensionality reduction for multi-class problems

We implement an assortment of various dimensionality reduction methods in order to evaluate if adequate lift in the ML performance is evident to justify dimensionality reduction. The dimensionality reduction methods implemented were various combinations of autoencoder frameworks (i.e., number of nodes and layers) and activation functions. Typical methods such as PCA were deemed unnecessary to implement since a single layer (SL) linear activation function is strongly comparable to PCA. To create a one-to-one comparison against the proposed NAPS framework, SMOTE was also implemented to these higher dimensionality problems to improve class imbalances.

Baseline Performance. Prior to implementing the dimensional reduction experiments, we obtained a baseline by simply examining the performance before we implemented any dimensionality reduction. Considering our concerns about how dimensionality reduction methods may actually hurt performance by capturing irrelevant information, we took two baselines: one experiment with strictly the raw data (SMOTE was not implemented) and the implementation of SMOTE with the Random Forest Algorithm (RF) with 150 bags. Table 6 is the confusion matrix and accuracy (A_c = 68.37%) for applying an RF algorithm to the raw data set. Utilizing RF and SMOTE together, Table 7, demonstrates a drop in performance accuracy. However, if you examine the confusion matrix in Table 6, the precision and recall is very poor for the other classes (θ_2, θ_3, θ_4). Thus, we can note how SMOTE improves performance for minority classes by adjusting the class imbalances.

Table 6
RF (A_c = 68.37%).

		Predicted			
		θ_1	θ_2	θ_3	θ_4
Actual	θ_1	**632**	24	6	1
	θ_2	93	**24**	1	0
	θ_3	124	5	**3**	0
	θ_4	43	2	0	**0**

Table 7
RF & SMOTE (A_c = 61.37%).

		Predicted			
		θ_1	θ_2	θ_3	θ_4
Actual	θ_1	**512**	71	68	12
	θ_2	59	**52**	5	2
	θ_3	95	13	**21**	3
	θ_4	32	6	4	**3**

Table 8
SL autoencoders.

Autoencoder size	Function	Accuracy	Function	Accuracy
70	logSig	55.63%	SatLin	54.07%
60	logSig	55.94%	SatLin	53.03%
50	logSig	57.51%	SatLin	51.57%
40	logSig	55.21%	SatLin	54.28%
32	logSig	55.63%	SatLin	50.42%
24	logSig	56.47%	SatLin	50.51%
16	logSig	56.36%	SatLin	50.93%
8	logSig	52.29%	SatLin	48.84%
5	logSig	48.23%	SatLin	42.49%

Table 9
SL autoencoder LogSig (A_c = 57.51%).

		Predicted			
		θ_1	θ_2	θ_3	θ_4
Actual	θ_1	**465**	79	82	37
	θ_2	49	**52**	12	5
	θ_3	74	16	**33**	9
	θ_4	27	3	14	**1**

Single Layer Autoencoders. To address this research question, we first started with a single-layer autoencoder framework, where we ran experiments using two different activation functions and nine different neural network (NN) node architectures. The random forest algorithm with 150 bags was implemented on these reduced feature sets from the autoencoders. Table 8, provides the averaged accuracy over 5-fold cross-validation for each NN architecture and activation function. These two activation functions essentially allowed us to compare how linear and non-linear manifolds may increase accuracy. We can note that across all the NN architectures (autoencoder size/nodes), the non-linear activation function (logsig) provides an increase in performance downstream. In Table 9, we provide the confusion matrix for autoencoder with 50 nodes that utilized the "logsig" activation function, which was then stacked with a random forest providing an accuracy of 57.51% for the four-class problem. However, when compared to Table 7, the most accurate autoencoder, which used the same RF paradigm, still performs the worst across all classes.

Table 10
Deep network architectures.

Layer 1 size	Layer 2 size	Function	Accuracy
220	8	logSig	51.77%
180	16	logSig	53.13%
100	32	logSig	54.27%
85	45	logSig	52.82%
120	64	logSig	53.44%

Table 11
Deep network ($A_c = 54.27\%$).

		Predicted			
		θ_1	θ_2	θ_3	θ_4
Actual	θ_1	**453**	72	107	31
	θ_2	58	**38**	16	6
	θ_3	73	22	**28**	9
	θ_4	29	8	7	**1**

Table 12
SL autocoders & RF all-vs-one.

Autoencoder size	Function	Accuracy	Function	Accuracy
185	logSig	61.17%	SatLin	55.63%
128	logSig	62.84%	SatLin	55.94%
64	logSig	62.84%	SatLin	57.51%
32	logSig	60.55%	SatLin	55.21%
16	logSig	57.82%	SatLin	55.63%
8	logSig	54.17%	SatLin	56.47%

Table 13
Single layer encoder (64 Vars.) Utilizing a random forest all-vs-one structure ($A_c = 62.84\%$).

		Predicted			
		θ_1	θ_2	θ_3	θ_4
Actual	θ_1	**522**	54	65	22
	θ_2	57	**44**	15	2
	θ_3	85	11	**33**	3
	θ_4	30	4	8	**3**

Table 14
Raw data set utilizing random forest all-vs-one structure ($A_c = 64.91\%$).

		Predicted			
		θ_1	θ_2	θ_3	θ_4
Actual	θ_1	**543**	57	53	10
	θ_2	59	**54**	2	3
	θ_3	96	14	**21**	1
	θ_4	34	4	6	**1**

Deep Networks. The dimensionality reduction experiments were then extended for designing two-layer autoencoders with a third soft layer used to train on the feature set. A simulation was done to test various combinations of the NN architecture, where combination of the first layer consisted of 220, 180, 120, 100, or 85 nodes, and the second layer consisted of combinations of 8, 16, 32, 45, or 64. This produced 25 different architectures. In Table 10, we provide the most accurate architecture for each second layer combination, where Table 11 is the confusion matrix for the highest performing architecture. As shown in the table, the current deep architectures are one of the lowest performing models.

RQ1 Takeaway. There was not a single dimensionality reduction method that provided a better lift in performance when compared to the baseline approach. We hypothesize that because of subject-to-subject variability within the experimental data, this causes anomalous patterns in the feature space, which cannot be generalized or clustered as relevant information through an unsupervised fashion [39]. Thus, when it comes to human performance data, it is innate for these unsupervised dimensionality reduction methods to capture irrelevant information for their new projections of the feature space.

6.2. RQ2: Dimensionality reduction combined with augmented response variables

In this subsection, we aim to address how the combination of simplified response variables would allow us to better handle the sparsity of the wide data for increased performance. We addressed this issue by applying upsampling using SMOTE, augmenting the response variable, and reducing the dimensionality (feature space). The augmented response variable approach we utilized is the all-vs-one approach. Thus, four-model binary models are designed, and we took the highest singleton probability as the predicted class. In Table 12, we once again examine the different NN encoder architectures and activation functions, where we achieved a tie for the best performance for a 128 and 64 node autoencoder using the nonlinear activation function. Note that there is an approximate 5% increase in performance compared to the previous method, which did not augment the response variable. In addition, the nonlinear activation function out-performed the linear activation. In Table 13, we provide the confusion matrix for the single layer 64 nodes autoencoder using the "logsig" activation function.

However, based on our findings with *RQ2* and our hypothesis from *RQ1* that dimensionality reduction is capturing irrelevant information for this human performance data. Thus, it behooved us to also remove the autoencoder from the ML design approach for further analysis. This is supported by Table 14 which demonstrates an increase in performance accuracy when compared to the other all-vs-one methods, Table 13. In addition, there is also a slight performance increase from the original baseline model using SMOTE, Table 7.

RQ2 Takeaway. Once again, we demonstrated that unsupervised dimensionality reduction decreased the performance. However, when strictly comparing the approach of augmenting the response variable (i.e., All-vs-One) to the original baseline in Table 7, we can note a slight increase in overall accuracy. Thus, by augmenting the response variable, we can aid in the uncertainty of the model by reducing the sparsity of the data set. However, the model performance is not optimal because of the large feature space the model is forced to cover.

6.3. RQ3: NAPS fusion approach

Baseline. From *RQ2*, we stated that we can improve the performance of the model if the feature space is reduced. We first examined a framework that uses the top-performing six random forest (RF) models (6 different subspaces in the feature space) that span the feature space. The predicted classes of each model are reported and tallied for a committee vote to determine the predicted class. In Table 15, we can note a significant increase in the predictive performance when compared to the models in *RQ1* and *RQ2*. Simply through deductive reasoning, we can attribute this performance increase to the small feature set and combination of features sets (sensors). This claim can be supported by examining how high dimensionality and low samples negatively alter classification performance and the SMOTE Algorithm. In order to increase the dimensionality of a classification problem, the number of training samples must increase as well. Otherwise, the classifier performance will decrease. When training samples are fixed, and dimensionality increases, the density of the training sample within the vector space will exponentially decrease. This lack of density in the feature space or sparsity imposes uncertainty on the model since we become uncertain as to whether the classification boundary set is correct. In addition, this affects how we attempt to increase the minor training samples by using SMOTE. SMOTE works on the nearest neighbor algorithm, and if our vector space is too sparse, the nearest neighbor algorithms begin to break down since the distances between

Table 15
4 class committee voting model ($A_c = 83.04\%$).

		Predicted			
		θ_1	θ_2	θ_3	θ_4
Actual	θ_1	**659**	1	0	0
	θ_2	74	**43**	1	0
	θ_3	49	6	**77**	0
	θ_4	31	3	9	**2**

the adjacent neighbors are pushed to new hyperplanes as the dimensionality increases with fixed sample size. Therefore, we are able to properly apply SMOTE to increase the sample size. However, utilizing a small subspace within the feature space (a sensor) should enhance SMOTE's capabilities.

NAPS Fusion Approach (Tripleton Set). Comparing the baseline classical approach in Table 15 which utilizes small models that strictly utilize the original response variable labels (a four-class problem, referred to as the X_1 response combination) that span the feature space, we demonstrated a significant lift in the predictive performance. However, utilizing the NAPS Fusion approach, we simplify the model's decision boundaries by augmenting the response variable into tripleton sets (i.e., X_2, X_3, X_4, and X_5). This all-vs-one approach augments the response variable, allowing for a more generalizable ML model to fit the data, which then can be later combined using fusion to examine the supports strictly for the four classes. In addition to this augmented response variable case (where we originally had high dimensionality, low samples, and class imbalance modeling issues), we instantaneously create more data by merging the responses together and reducing the sparsity vector space, allowing us to use SMOTE for improved model performance as well.

The NAPS Fusion approach's sensor selection paradigm is critical since we randomly generated 175 models for each combination of responses variables (a total of 7 different combinations variations X_2 to X_8). Thus, we had 1225 models to choose from, but we only used 24 of them for these reported results. From each response combination, X_p, we picked the best six models with the lowest uncertainty. Thus, the NAPS Fusion approach combines only the best six sensors for each X_p, which then fuses these models together, shown in Table 16. Once again, we note an even larger significant improvement from the baseline committee voting approach (from 83.04% to 95.29% accuracy, as well as the precision and recall) that is compared to using the best six models within the X_1 original response variable labels. This demonstrates the power that the NAPS Fusion framework has to improve the classification performance through smaller models and augmented response variables.

NAPS Fusion Approach (Doubleton Set). We now approach the problem from the perspective that we have ambiguous class labels, leaving us ignorant of the class label's "true" ground truth. We utilize the NAPS Fusion approach to address the augmented response variable for doubleton sets (i.e., X_6, X_7, and X_8). Typically, we are unable to handle ignorance when we have conflicting class labels. However, our proposed DS framework can handle this conflict and ambiguous labels. Similarly, we use the best six models that each X_p produces within the double sets, yielding 18 models to fuse. In Table 17, we demonstrate still a very high-performance accuracy of $A_c = 93.93\%$ when multiple class doubleton class labels are formed to an ML paradigm and fused together. This is an extremely critical aspect of human–machine interaction research, especially when class labels taken from annotations formed by subject-matter experts are in disagreement. Thus, if one subject-matter expert labels the event as θ_1 and another subject-matter expert labels the event as θ_2, we can set the class label to train as a doubleton set of $\{\theta_1, \theta_2\}$. This enables us approach the ML framework to combat issues of mixed states or state transitions within the data set.

RQ3 Takeaway. The NAPS Fusion framework provides a superior lift in performance compared to other approaches. To elucidate on the performance gains of NAPS Fusion, let us discuss how traditional methods lack the key framework infrastructure that restricts the performance potential of models within human data sets. This reduction of performance is driven by an *insufficient data structure* and an *inconsistent statistical structure*, which increases the uncertainty of traditional models, leading to poor classification. Through this discussion we highlight how NAPS Fusion overcomes these limitations to improve classification performance.

An insufficient data structure occurs when we have the combination of three major data limitations present within a predictive modeling environment [40]. These three data limitations within the data structure are: (1) high dimensionality of features which drives the feature space to be sparse and degrades the effectiveness to measure dissimilarity between samples; (2) low number of samples widen the confidence interval of the estimated model parameter's which increases the uncertainty of the decision boundary; and (3) class imbalances alter the prior of the model's probability. When *only one* of these data limitations is introduced there are several classical ways of manipulating the data structure to produce a more sufficient data set for the predictive model to learn. For example, when the issue of class imbalances is only present within the data structure, we can adjust the number of samples based on their class by re-sampling (down-sampling or up-sampling) to correct the model's biasing to avoid an inappropriate prior probability [41]. However, when an additional limitation enters the data structure, the statistical ability to train the classifier degrades from the increase of unhandled uncertainty entering into the predictive paradigm. For instance, during a down-sampling paradigm with low samples, the reduction of additional samples (i.e., observations) leads to wider confidence intervals of the estimated model parameters which define the decision boundary. This creates a flexible decision boundary that has a high uncertainty, causing inaccurate model classification [28,42]. From an up-sampling perspective with high dimensionality and low samples, as we can apply SMOTE a nearest neighbor approach to improve the model bias, the high dimensionality deteriorates the distance measurements for properly assessing similarity [43]. The deterioration of the distance measurements is related to the "curse of dimensionality" where, as dimensionality increases, the distance measurements start to lose their meaning and effectiveness, increasing the uncertainty for decision boundary [44–46]. This creates inaccurate placement for new synthetic samples, thereby impacting the learning for the classifier downstream. In conclusion, as the classifier is introduced to more data structure limitations the potential for an accurate and precise prediction becomes more unattainable.

With respect to insufficient data structures, NAPS Fusion framework attenuates or potentially eliminates compounding negative implications caused by targeting the limitations using DS theory. We first address the high dimensionality of features by creating a multitude of small models (e.g., small set of predictive features) that span the feature space. By creating models from subsets that randomly span the original data set, we convert a single high-dimensional problem into multiple low-dimensional problems. The decisions resulting from the smaller models are later fused by using DS theory to incorporate the additional predictive variables from adjacent subspaces (discussed in Section 3.2.2). Through the use of smaller generalized models, we can form tighter confidence intervals (i.e., reduced uncertainty) for the estimated model parameters for an improved decision boundary. The second way DS theory is leveraged to reduce uncertainty for the classifier is by utilizing the property of ignorance. By exploiting the DS notion of ignorance, we can merge classes into augmented response variables, which provide more samples for training for a specific class (refer to Section 3.3). The augmentation of the response variables also enables the model to not be required to fit multiple classes, thus providing more generalizable decision boundaries. This allows us to properly implement data upsampling techniques, such as SMOTE, for every model to resolve any remaining class imbalances at a lower dimensional feature space.

Through the leveraging of these two DS theory properties, we apply DS fusion to these small models that span the feature space to mimic a high dimensionality through DS fusion. Due to the use of augmented response variables, each model has an improved confidence interval of the model's estimated decision boundary (i.e., reduced model uncertainty) for enhanced classification performance. It is important to note that at this point within the model we are only leveraging the DS framework to improve the uncertainty within the ML classifier by augmenting the data structure. This is independent to how the DS framework manages uncertainty and ignorance to improve the mass allocation of its propositions which is later implemented.

The other modeling limitations stem from an inconsistent statistical structure within the human performance data. An inconsistent statistical structure can occur when a particular set of samples do not follow the same statistical boundary requirement of another set of samples within the same dataset. However, both sets are linked to the same class label. This occurs habitually within human subject data, where the subject-to-subject variability greatly differs, preventing a generalized predictive model to be deployed for the entire dataset. For example, as depicted in Fig. 6, this can lead to an appropriate classification of the sample for subject 1 when model 1 is deployed, but a misclassification when model 1 is deployed on subject 2. Likewise, the inverse scenario can occur when model 2 is applied to both subjects. More specifically, the example in the figure demonstrates how two different subjects can experience the same stimulus, like hypoxia (e.g., a change in their environment such as a delivery of a drug, a stressor, etc.), but physiologically react differently. The different responses to a stimulus across subjects can lead to uncertain model decision boundaries from subject to subject. Therefore, no single classical model can be generalized to the entire set of subjects. The issue of inconsistent statistical structures within the data is further exacerbated by the high dimensionality of the data. Commonly, we deploy dimensionality reduction methods on the data like autoencoders and principal component analysis (PCA) to learn more complex projections of the data at a lower dimensional feature space. Despite their widespread use, dimensionality reduction methods like PCA and autoencoders do not always map to a lower dimensional space appropriately. For example in PCA, higher-order principal components can mainly consist of noise or irrelevant information [39]. This tends to occur when the features that are required to be extracted are from a minority class (i.e., subset of subjects responds a specific way) and cannot be extracted from maximizing the variance [47,48]. Within literature, this tends to occur when PCA maximizes the variance of the features of the majority class, causing the minority class to be neglected [47,48]. This translation between minor and majority class is analogous to our study when subject-to-subject variability occurs within the data, where specific subsets of subjects respond differently to a stimulus. Thus, a subset of subjects will have improper mappings for their principles components, because the PCA maximization of the variance was biased to the more popular response to the stimuli (e.g., similar to the majority issue in literature). Similarly, autoencoders are created with the sole purpose of compressing data into a reduced latent space, while disregarding information on the variability of the data set. Therefore, when trained on wide data sets with significant subject-to-subject variability, autoencoders create a static model that scales poorly on subsets of the data set that exhibit a high degree of variability. In essence, the autocoder captures the "average" of the dataset such that it performs well on the entire dataset [49]. This explains why the use of dimensionality reduction methods, like autoencoders or PCA, within our experimental data greatly underperforms when compared to the findings solely utilizing the raw data. This can be seen when comparing the raw data performance in Tables 14 and 15 to the autoencoder performance in Table 13.

To address subject-to-subject variability and high data dimensionality, NAPS Fusion allows us to span the feature space at a higher dimension by fusing multiple small models, with the major caveat that the framework will naively adapt to each specific sample (i.e., adapting to a subject's stimulus response). This naive adaptation accounts for the subject-to-subject variability within the full dataset through a dynamic weighting of models based on the uncertainty assignment (discussed in Section 4.3). The models with higher uncertainty are weighted less by NAPS Fusion for certain samples/subjects and weighted higher for models with lower uncertainty, which naively accounts for subject-to-subject variability. This explains the additional increases in classification performance for NAPS Fusion when compared to the committee voting classification paradigm in Tables 15 and 16.

Table 16
NAPS fusion model ($A_c = 95.29\%$) tripleton set or all-vs-one response combination.

		Predicted			
		θ_1	θ_2	θ_3	θ_4
Actual	θ_1	**658**	1	1	0
	θ_2	0	**118**	0	0
	θ_3	0	5	**127**	0
	θ_4	23	3	12	**7**

Table 17
NAPS fusion utilizing ambiguous class labels ($A_c = 93.93\%$ Doubleton Set).

		Predicted			
		θ_1	θ_2	θ_3	θ_4
Actual	θ_1	**659**	1	1	0
	θ_2	1	**117**	0	0
	θ_3	5	10	**116**	1
	θ_4	23	10	7	**5**

NAPS Fusion opens up the discussion of various methods that could be implemented to improve the model selection, fusion of models, and dimensionality reduction of wide data sets. As we know, there is no "free lunch" when it comes to machine learning. The computational complexity of the DS framework becomes exponentially higher as the number of classes increase. However, the need to implement a DS fusion framework not only demonstrates superior classification performance, but also allows us to properly manage ambiguities in the class label.

7. Conclusion

Many cognitive performance experiments for human–machine interaction produce a small sample size, have large class imbalances, and have a high dimensionality feature space. Typically, we are faced with machine learning problems that have one or two of these data limitations simultaneously. However, when all three of these constraints occur, the standard approaches for handling each problem individually essentially fail. For instance, class imbalances could be handled by down-sampling our data to adjust for imbalance in the classes, but we do not have enough data for down-sampling. Conversely, we can up-sample by using SMOTE, but as we discussed and demonstrated, the benefits of SMOTE degrade when we have a low amount of samples and high dimensionality. In addition, we can try to address the problem by applying dimensionality reduction methods, but they are unable to capture relevant information, most likely due to the small sample size, class imbalances, and subject-to-subject variability.

The NAPS Fusion framework addresses how we can overcome these experimental data issues and ML challenges. NAPS accomplishes this through appropriate model selection, uncertainty assignment, augmentation of the response variable, and fusion of sensors that span the feature space that allow SMOTE not only to adjust for original class imbalances but also for the imbalances created by the augmented response variables. The fusion of the models is done at each sample, and the uncertainty calculation therefore adapts and weights the impact of the output of the model across the top 6 models within X_P and X_{P+k} response combinations. The work presented utilizes

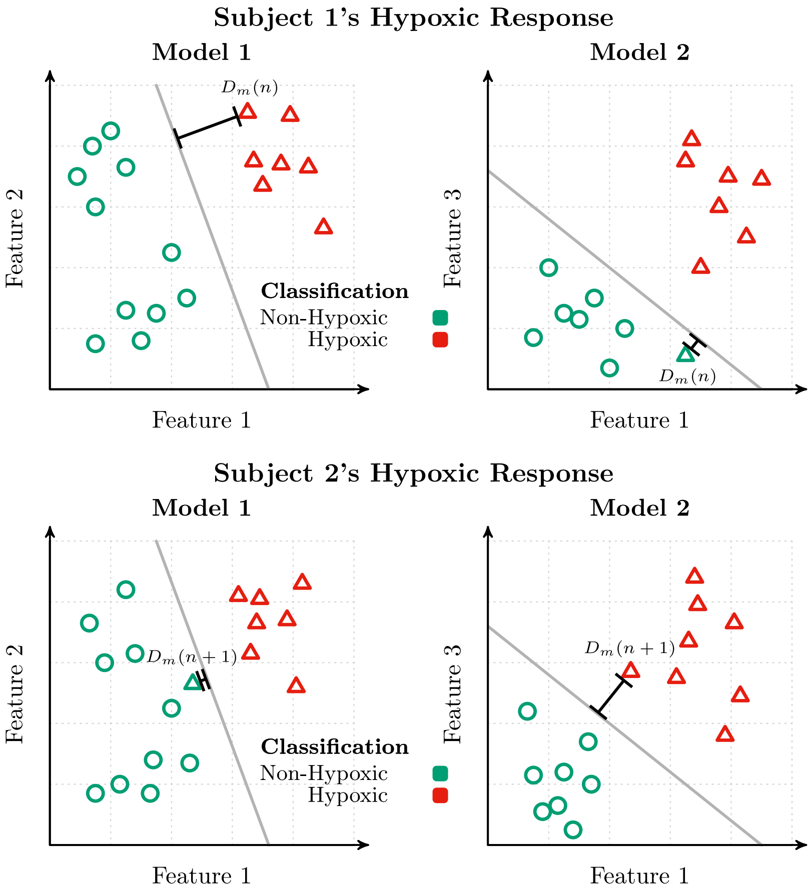


Fig. 6. The condition responses for two subjects using two distinct bagged models are shown here. The averaged decision boundary is shown as a gray line and the distance from sample $D_m(n)$ to the decision boundary is shown in black. The ground truth labels are denoted by circles (non-hypoxic) and triangles (hypoxic), while the classification labels are denoted by the colors green (non-hypoxic) and red (hypoxic). A mismatch in the color and shape of the label (subject 1, model 2 and subject 2, model 1) examplifies a misclassification of a data sample driven by subject-to-subject variability. In addition to this figure demonstrating subject-to-subject variablity, it also visually describes how a model is weighted differently across samples, $D_m(n)$ and $D_m(n+1)$. A bagged model's inability to clearly define the votes of the winning class will create a higher uncertainty for these specific samples (more on this in Section 4.3). These particular samples of higher uncertainty will tend to be closer to the decision boundary. Therefore, our uncertainty is an indirect measurement of the distance of the samples to the averaged decision boundary of the bagged model. This uncertainty measurement will alter the weights of the models with highly conflicting votes that are potential weak predictors for a specific subject's response to their stimuli. For example, when predicting hypoxia for subject 2, NAPS Fusion weights model 1 significantly less than model 2 to reduce the impact of models 1's missclassifcation (i.e., high conflict). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

DCR as a means of fusion and is one of the most rudimentary fusion algorithms, which opens up research areas for exploring methods for more optimal model fusion paradigms, and model selection methods. This DS approach for combining models under NAPS opens the door for cognitive performance research to move forward into open-world environments, where class label conflict occurs frequently. Further, it could potentially reduce costs associated with experimental human data collection. Moreover, this framework provides an extremely modular design, where new models and hardware modalities can be interchanged easily without tuning and training from ground zero, enabling new data sets to be amended to an original study design.

CRediT authorship contribution statement

Nicholas J. Napoli: Conceptualization, Visualization, Investigation, Validation, Formal analysis, Methodology, Software. **Chad L. Stephens:** Data curation, Visualization, Investigation, Resources, Supervision, Project administration. **Kellie D. Kennedy:** Data curation, Investigation. **Laura E. Barnes:** Writing – review & editing. **Ezequiel Juarez Garcia:** Software, Investigation, Writing – review & editing, Visualization, Data curation. **Angela R. Harrivel:** Project administration, Review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

This work was supported by the National Institute of Aerospace and NASA Langley Research Center, as well as the NIH's NHLBI under grant *L30 HL159818: Modeling of Breathing Dynamics to Discriminate ALS Ventilatory Compensation and Failure*.

References

[1] A.R. Harrivel, C.L. Stephens, R.J. Milletich, C.M. Heinich, M.C. Last, N.J. Napoli, N. Abraham, L.J. Prinzel, M.A. Motter, A.T. Pope, Prediction of cognitive states during flight simulation using multimodal psychophysiological sensing, in: AIAA Information Systems-AIAA Infotech @ Aerospace, American Institute of Aeronautics and Astronautics, 2017.

[2] Y. Vaizman, K. Ellis, G. Lanckriet, Recognizing detailed human context in the wild from smartphones and smartwatches, IEEE Pervasive Comput. 16 (4) (2017) 62–74.

[3] C. Wu, H. Fritz, S. Bastami, J.P. Maestre, E. Thomaz, C. Julien, D.M. Castelli, K. de Barbaro, S.K. Bearman, G.M. Harari, R. Cameron Craddock, K.A. Kinney, S.D. Gosling, D.M. Schnyer, Z. Nagy, Multi-modal data collection for measuring health, behavior, and living environment of large-scale participant cohorts, GigaScience 10 (6) (2021).

[4] J. Wang, Z. Wang, S. Qiu, J. Xu, H. Zhao, G. Fortino, M. Habib, A selection framework of sensor combination feature subset for human motion phase segmentation, Inf. Fusion 70 (2021) 1–11.

[5] N.J. Napoli, S. Adams, A.R. Harrivel, C. Stephens, K. Kennedy, M. Paliwal, W. Scherer, Exploring cognitive states: Temporal methods for detecting and characterizing physiological fingerprints, in: AIAA SciTech Forum, 2020, p. 1193.

[6] J.M. Henderson, S.V. Shinkareva, J. Wang, S.G. Luke, J. Olejarczyk, Predicting cognitive state from eye movements, PLOS ONE 8 (2013) 1–6.

[7] C. Wu, A.N. Barczyk, R.C. Craddock, G.M. Harari, E. Thomaz, J.D. Shumake, C.G. Beevers, S.D. Gosling, D.M. Schnyer, Improving prediction of real-time loneliness and companionship type using geosocial features of personal smartphone data, Smart Health 20 (2021) 100180.

[8] J.A. Russell, A circumplex model of affect, J. Personal. Soc. Psychol. 39 (6) (1980) 1161–1178.

[9] J.N. Bailenson, E.D. Pontikakis, I.B. Mauss, J.J. Gross, M.E. Jabon, C.A. Hutcherson, C. Nass, O. John, Real-time classification of evoked emotions using facial feature tracking and physiological responses, Int. J. Human-Comput. Stud. 66 (5) (2008) 303–317.

[10] B. Cowley, M. Filetti, K. Lukander, J. Torniainen, A. Henelius, L. Ahonen, O. Barral, I. Kosunen, T. Valtonen, M. Huotilainen, N. Ravaja, G. Jacucci, The Psychophysiology Primer: A guide to methods and a broad review with a focus on human-computer interaction foundations and trends in human-computer interaction, Vol. 9, Now Foundations and Trends, 2016, pp. 150–307.

[11] R.L. Mandryk, M.S. Atkins, A fuzzy physiological approach for continuously modeling emotion during interaction with play technologies, Int. J. Human-Comput. Stud. 65 (4) (2007) 329–347.

[12] R. Haenni, S. Hartmann, Modeling partially reliable information sources: A general approach based on Dempster-Shafer theory, Inf. Fusion 7 (4) (2006) 361–379.

[13] J. Wang, Y. Hu, F. Xiao, X. Deng, Y. Deng, A novel method to use fuzzy soft sets in decision making based on ambiguity measure and Dempster-Shafer theory of evidence: An application in medical diagnosis, Artif. Intell. Med. 69 (2016) 1–11.

[14] E. Lefevre, P. Vannorenberghe, O. Colot, Using information criteria in Dempster-Shafer basic belief assignment, in: IEEE International Fuzzy Systems Conference Proceedings (I), 1999, pp. 173–178.

[15] R.R. Yager, L. Liu, Classic Works of Dempster-Shafer of Belief Functions, Springer, 2008.

[16] A. Gelman, The boxer, the wrestler and the coin flip: A paradox of robust Bayesian inference and belief functions, Amer. Statist. 60 (2) (2006) 146–150.

[17] G. Fioretti, A mathematical theory of evidence for G.L.S. Shackle, Mind Soc. 2 (2001) 77–97.

[18] K. Sentz, S. Ferson, Combination of Evidence in Dempster-Shafter Theory (Ph.D. thesis), Binghamton University, P.O. Box 6000 Binghamton, NY 13902-6000, 2002.

[19] N. Napoli, K. Leach, L. Barnes, W. Weimer, A MapReduce framework to improve template matching uncertainty, in: 2016 International Conference on Big Data and Smart Computing (BigComp), 2016, pp. 77–84.

[20] N.J. Napoli, A. Harrivel, A. Raz, Improving physiological monitoring sensor systems for pilots, Aerospace Am. 12 (2020).

[21] C. Stephens, K. Kennedy, N. Napoli, M. Demas, L. Barnes, B. Crook, R. Williams, M. Carolyn Last, P. Schutte, Effects on task performance and psychophysiological measures of performance during normobaric hypoxia exposure, in: 19th International Symposium on Aviation Psychology, 2017, p. 202.

[22] F. Petrassi, P. Hodkinson, P. Walters, S. Gaydos, Hypoxic hypoxia at moderate altitudes: Review of the state of the science, Aviation Space Environ. Med. 83 (10) (2012) 975–984.

[23] N.J. Napoli, M. Demas, C.L. Stephens, K.D. Kennedy, A.R. Harrivel, L.E. Barnes, A.T. Pope, Activation complexity: A cognitive impairment tool for characterizing neuro-isolation, Sci. Rep. 10 (1) (2020) 1–20.

[24] T. Halverson, B. Reynolds, L. Blaha, SIMCog-JS: Simplified interfacing for modeling cognition - JavaScript, in: Proceedings of the International Conference on Cognitive Modeling, 2015, pp. 39–44.

[25] N. Napoli, Characterizing Uncertainty in Sensor Fusion to Improve Predictive Models, Online Archive of University of Virginia, 2018, pp. 1–201.

[26] Y. Santiago-Espada, R.R. Myer, K. A. Latorella, J.R. Comstock, The multi-attribute task battery II (MATB-II) software for human performance and workload research: A user's guide, in: NASA/TM-2011-217164, 2011.

[27] N.J. Napoli, M.W. Demas, S. Mendu, C.L. Stephens, K.D. Kennedy, A.R. Harrivel, R.E. Bailey, L.E. Barnes, Uncertainty in heart rate complexity metrics caused by R-peak perturbations, Comput. Biol. Med. 103 (2018) 198—207.

[28] N.J. Napoli, W. Barnhardt, M.E. Kotoriy, J.S. Young, L.E. Barnes, Relative mortality analysis: A new tool to evaluate clinical performance in trauma centers, IISE Trans. Healthcare Syst. Eng. 7 (3) (2017) 181–191.

[29] R.B. Kline, Principles and Practice of Structural Equation Modeling, third ed., Guilford publications, 2015.

[30] P.M. Bentler, C.P. Chou, Practical issues in structural modeling, Sociol. Methods Res. 16 (1) (1987) 78–117.

[31] J.B. Schreiber, A. Nora, F.K. Stage, E.A. Barlow, J. King, Reporting structural equation modeling and confirmatory factor analysis results: A review, J. Educ. Res. 99 (6) (2005) 323–338.

[32] N.V. Chawla, K.W. Bowyer, L.O. Hall, W.P. Kegelmeyer, SMOTE: Synthetic minority over-sampling technique, J. Artificial Intelligence Res. 16 (2002) 321–357.

[33] H. He, Y. Bai, E.A. Garcia, S. Li, ADASYN: Adaptive synthetic sampling approach for imbalanced learning, in: 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence), 2008, pp. 1322–1328.

[34] H. He, E.A. Garcia, Learning from imbalanced data, IEEE Trans. Knowl. Data Eng. 21 (9) (2009) 1263–1284.

[35] G. Shafer, A Mathematical Theory of Evidence, Princeton University Press, Princeton, NJ, 1976.

[36] I. Davidson, W. Fan, When efficient model averaging out-performs boosting and bagging, in: J. Fürnkranz, T. Scheffer, M. Spiliopoulou (Eds.), Knowledge Discovery in Databases: PKDD 2006: 10th European Conference on Principles and Practice of Knowledge Discovery in Databases Berlin, Germany, September 18-22, 2006 Proceedings, Springer Berlin Heidelberg, Berlin, Heidelberg, 2006, pp. 478–486.

[37] M. Galar, A. Fernández, E. Barrenechea, H. Bustince, F. Herrera, An overview of ensemble methods for binary classifiers in multi-class problems: Experimental study on one-vs-one and one-vs-all schemes, Pattern Recognit. 44 (8) (2011) 1761–1776.

[38] C. Bishop, Pattern Recognition and Machine Learning, Springer, 2006.

[39] M. Lenz, F.-J. Muller, M. Zenke, A. Schuppert, Principal components analysis and the reported low intrinsic dimensionality of gene expression microarray data, Sci. Rep. (2016) 1–11.

[40] L. Shen, M.J. Er, W. Liu, Y. Fan, Q. Yin, Population structure-learned classifier for high-dimension low-sample-size class-imbalanced problem, Eng. Appl. Artif. Intell. 111 (2022) 104828.

[41] G. Shmueli, To explain or to predict? Statist. Sci. (3) (2010).

[42] V. Bugera, S. Uryasev, G. Zrazhevsky, Classification using optimization: Application to credit ratings of bonds, in: Computational Methods in Financial Engineering, Springer, 2008, pp. 211–237.

[43] Learning distance to subspace for the nearest subspace methods in high-dimensional data classification, Inform. Sci. 481 (2019) 69–80.

[44] D. Foley, Considerations of sample and feature size, IEEE Trans. Inform. Theory 18 (5) (1972) 618–626.

[45] G.V. Trunk, A problem of dimensionality: A simple example, IEEE Trans. Pattern Anal. Mach. Intell. (3) (1979) 306–307.

[46] C. Beleites, U. Neugebauer, T. Bocklitz, C. Krafft, J. Popp, Sample size planning for classification models, Anal. Chim. Acta 760 (2013) 25–33.

[47] C. Grigo, P.-S. Koutsourelakis, Bayesian model and dimension reduction for uncertainty propagation: applications in random media, SIAM/ASA J. Uncertain. Quantif. 7 (1) (2019) 292–323.

[48] T.M. Padmaja, B.S. Raju, R.N. Hota, P.R. Krishna, Class imbalance and its effect on PCA preprocessing, Int. J. Knowl. Eng. Soft Data Paradigm 4 (3) (2014) 272–294.

[49] A. Maheshwari, P. Mitra, B. Sharma, Autoencoder: Issues, challenges and future prospect, in: Recent Innovations in Mechanical Engineering, Springer, 2022, pp. 257–266.