Transfer Function to Predict Warhead Fragmentation In-Flight Behavior from Static Data

Omkar S. Mulekar* and Riccardo Bevilacqua†
University of Florida, Gainesville, Florida 32611
Elisabetta L. Jerome‡
Air Force Test Center, Eglin Air Force Base, Florida 32542
and
Thomas J. Hatch-Aguilar§
U.S. Naval Air Warfare Center, China Lake, California 93555

https://doi.org/10.2514/1.J060226

In recent years, static warhead arena tests have used stereoscopic cameras to count fragments in real time and correlate their individual velocity and mass. This new technique provides the ability to generate higher confidence dynamic fragmentation data that in turn can be used for code validation and much more realistic lethality and collateral damage calculations. Arena tests involve detonating static warheads; whereas, in reality, warheads arrive and detonate at high speeds. State-of-the-art simulations for high-speed warhead detonations can be challenging and time consuming while potentially missing the relevant physics of real-world detonations. In this investigation, a framework to predict warhead fragment track characteristics from real-world static arena experimental data and dynamic simulation data is explored. A model is trained on simulation and experimental data to predict the number of fragments that pass through a defined surface of interest given warhead in-flight terminal conditions. Distributions of fragment–surface intersections are modeled by Gaussian mixture models (GMMs), and random forest regressors are trained to predict these GMMs. Monte Carlo methods are used to show that random forests trained by both simulation and experimental data can predict fragment–surface intersection distributions of both static and dynamic high-speed warhead configurations.

Nomenclature

f_i	=	individual regression tree in a random forest regressor
g	=	random forest regressor
$\mathcal{N}$	=	the Gaussian distribution, either single variate or multivariate
N	=	number of fragments
p	=	probability distribution described by a Gaussian mixture model
S	=	region on a sphere surface
$\boldsymbol{u}$	=	vector input to a random forest regressor or regression tree
x	=	scalar random variable in a single-variate distribution
$\boldsymbol{x}$	=	vector random variable in a multivariate distribution
$\boldsymbol{\Theta}$	=	parameters defining a probability distribution
θ	=	azimuthal angle
μ	=	mean of single-variate distribution
$\boldsymbol{\mu}$	=	mean vector of a multivariate distribution
π_k	=	mixing coefficient associated with components of a Gaussian mixture model
Σ	=	covariance matrix of a multivariate distribution
σ^2	=	variance of a single-variate distribution
ϕ	=	polar angle

Received 5 October 2020; revision received 5 April 2021; accepted for publication 9 April 2021; published online 8 July 2021.

*Ph.D. Student, Department of Mechanical and Aerospace Engineering, 939 Sweetwater Dr., MAE-A 211.
†Associate Professor, Department of Mechanical and Aerospace Engineering, 939 Sweetwater Dr., MAE-A 211.
‡Technical Advisor for Weapons T&E, 101 West D Ave., Bldg. 1, Suite 115.
§Weapons Leathality Analyst, Weapons Division, 1 Admin Circle.

I. Introduction

WARHEAD detonations eject fragments over large distances, causing collateral damage to structures, vehicles, and personnel. It is therefore imperative to accurately understand warhead detonation characteristics so that collateral damage estimates and lethality estimates can be used to make decisions regarding effective target application [1,2]. Many techniques exist to estimate the collateral damage and lethality of a fragmenting munition [3,4]. The fragmentation properties used to characterize lethality and collateral damage estimation include fragment counts, masses, velocities, and shapes [5]. In recent years, stereoscopic vision systems have been implemented in static arena tests to track individual fragments after a warhead detonation, providing data on fragment counts, masses, and velocities [6]. Although these experiments provide valuable, realistic information that improves lethality and collateral damage estimations, real warheads arrive at targets at high speeds [7]. Conversely, although simulation techniques allow for fragment track generation for nonstatic cases, there are many real-world effects (e.g., air resistance) not present [8].

Before 1943, much of the focus in ballistics research and development was on launch and flight systems, and not on the final delivery components. Analytical methods to model explosive fragmentation such as Mott's methods for exploding cylindrical shells were developed in the 1940s, and they are statistical in nature [9]. Munitions designers continue to use these analytical techniques to make mass distribution and velocity distribution predictions. As a consequence, Mott's theory has dominated empirical models in many fields (fragmentation, hypervelocity impacts, etc.). Similarly, the theories of the early to mid-20th century inform the simulation capabilities available at present day [10].

Several simulation tools have been developed to predict fragmentation and track behaviors. Picatinny arsenal fragmentation is one example that combines analytical and experimental techniques through high-strain/high-strain-rate computer modeling [5,11]. One software titled "Combined Hydro and Radiation Transport Diffusion Squared to the Three Halves", which was developed at Sandia

National Laboratories, uses Eulerian finite difference techniques to model multidimensional shock wave physics [12]. Simulation techniques have also been developed at NASA Johnson Space Center to model fragmentation from explosions and collisions of rocket bodies and spacecraft in low Earth orbit. These simulations, however, are largely based on empirical data; and they are not driven by physics-based equations [13]. Alternatively, a research software developed at the U.S. Naval Air Warfare Center's Weapons Division (NAWCWD), based on an iterative solution to Langevin's equations of motion, is leveraged in this paper because it captures multiphysics solutions found in the real world [14]. The NAWCWD research software solves the static and dynamic fragment flyouts, accounting for the multiphysics encountered in the trajectories emanating from an explosion. Additionally, the research software optimizes the simulations by leveraging an explicit vector operator partitioning on a Lagrangian frame of reference. Currently, the NAWCWD high-fidelity research software, which is currently under review by the U.S. Navy, represents one of the best methods to model static and dynamic fragment flyout from explosive detonations.

The *Joint Munitions Effectiveness Manual* provides a procedure for the collection of fragmentation data from static arena warhead detonations [15]. This manual has been used to characterize dynamic events during experiments using collection panels as well [16]. Also, high-speed optical stereoscopic video (HSV) techniques have been implemented for three-dimensional tracking of fragments after static detonations [17,18].

The goal of this research paper is to combine the benefits of real-world static detonation data with the benefits of machine learning techniques trained or grounded by dynamic simulation data from the NAWCWD fragment flyout research software. In effect, this becomes a novel method to accurately characterize warhead fragmentation distributions in the dynamic case. The end product is an equivalent transfer function that can predict in-flight fragment behavior, even though experimental data may only be available for the static case. Whereas data from static arena detonations capture the effects of real-world physics, simulation data allow the model to learn some of the physical effects derived from high-speed detonation scenarios. A model trained on a balanced combination of experimental and simulation data learns to incorporate the physical effects present in static experimental data along with the predictions for the dynamic cases for which only simulation data are available. The trained model can estimate high-speed detonation fragment flyouts for those scenarios where only limited experimental data are available. Ultimately, validation of the proposed model, along with the techniques presented in this paper, will be performed as dynamic warhead detonation experiments become a reality and experimental data for the dynamic case are available. This validation is a potential future work that cannot be done now since the dynamic case experimental data are not yet available.

This paper first gives an overview of the data available from static arena tests and simulations in Sec. II. Then, an overview of the machine learning techniques used, including Gaussian mixture models (GMMs) and random forest regressors, is given in Sec. III. The problem statement, the proposed model architecture, and the process for generating training data are given in Sec. IV. Monte Carlo methods for evaluating the performance of the model are given in Sec. V. Section VI presents the results of these Monte Carlo numerical evaluations. The results of assimilating experimental and simulation data into the random forest training set are discussed in Sec. VII. Future work is discussed in Sec. VIII, and conclusions are given in Sec. IX.

II. Description of Available Data

Data are available for this research effort in two primary forms: experimental data and simulation data. The experimental data come from static pipe bomb detonations performed at NAWCWD at China Lake. Similarly, the simulation data come from a research software that is developed and maintained by the NAWCWD at China Lake.

The experimental data from the static pipe bomb detonations are composed of 2, 5, and 7 mm ball-bearing tracks recorded from a pipe bomb detonation experiment. A stereoscopic vision system is used to perform fragment tracking, and so ball-bearing track data are available on one side of the detonation. Photographs from the experimental setup are shown in Fig. 1. An example of these ball-bearing tracks is in Fig. 2. Due to the radially symmetric placement of ball bearings around the pipe bomb, an assumption of symmetry is made on the experimental data available. The tracks shown on the $-Y$ side of the detonation are assumed to have symmetric pairs on the opposite side on the $+Y$ side. Every fragment position $[x, y, z]$ available in the data is duplicated to the position $[x, -y, z]$.

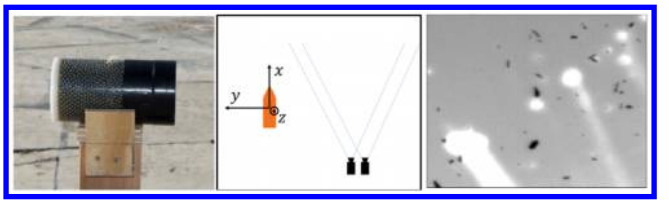

Fig. 1 Experimental setup for pipe bomb with 5 mm ball bearings to be tracked through stereoscopic HSV.

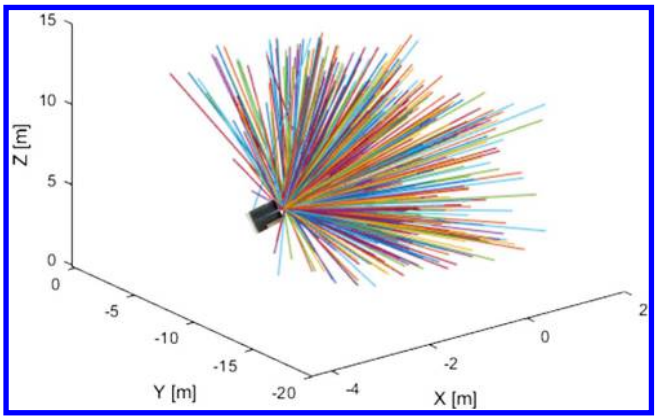

Fig. 2 5 mm ball-bearing tracks from static detonation experimental dataset.

The simulations iteratively solve for the Langevin's equations for each fragment trajectory. The physics includes drag and lift models as well as atmospherically realistic and community validated effects. The simulations provide comprehensive track data of impelled ball bearings for a variety of detonation attitudes and speeds. For the experimental cases, the data-collecting instruments used are limited in the number of recorded fragment tracks produced. In the simulation cases, however, tracks for all of the 3861 simulated fragments are available. The goal of this paper is to focus on leveraging these two datasets (experimental and simulation) and derive a complementary Machine Learning (ML) model that captures anything in between and expands on available applications. The simulation data are available for a range of static and dynamic cases with varying terminal speeds and system orientations at detonation. The delivered simulation cases are shown in Table 1, and a diagram of the coordinate system used in the simulations is shown in Fig. 3. The full factorial design of the implemented cases resulted in 1100 total simulations. In each simulation, a full time history of each fragment was generated, resulting in 1.31 TB of data.

Table 1 Table of NAWCWD simulation cases

Variables		Simulation cases	Number of cases	Units
Terminal speed	Magnitude	0, 152, 304, 457, 609, 762, 914, 1066, 1219, 1371, 1524	11	m/s
Orientation of weapon at burst point	Pitch	−90, −60, −30, 0	4	degrees
	Yaw	−60, −30, 0, 30, 60	5	
	Roll	0, 45, 90, 135, 180	5	

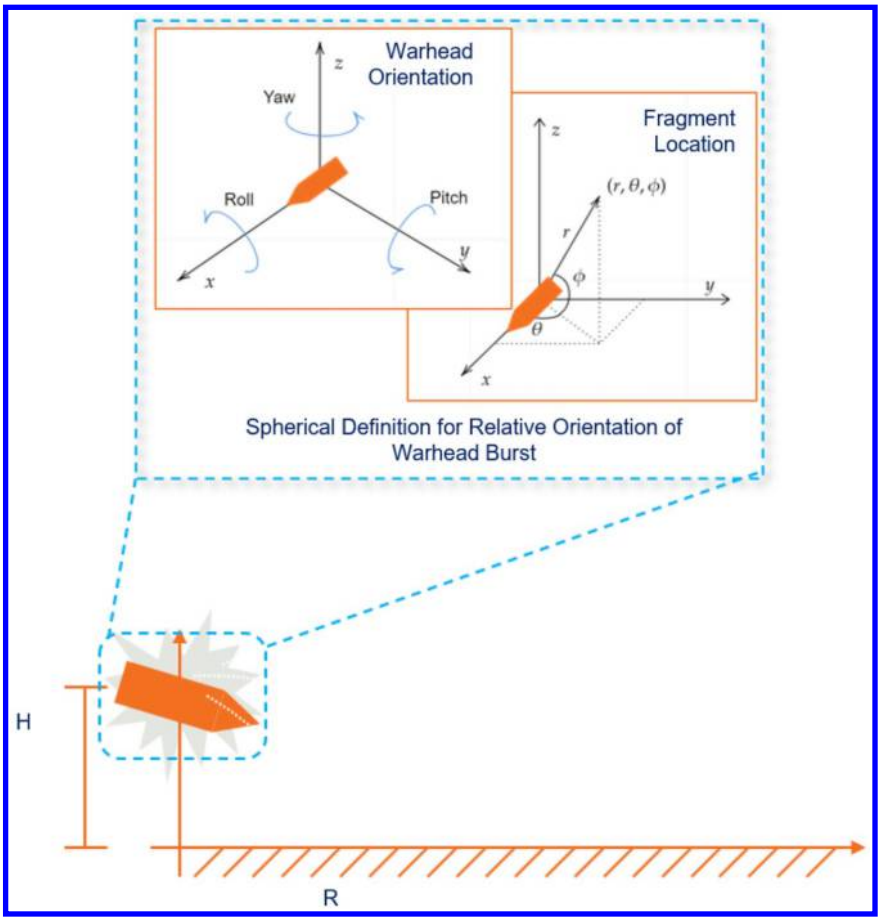

Fig. 3 Diagram of coordinate system used in simulations.

As a comment on uncertainty in the input data, and consequently the model, this research focuses only on using a prototype ML system trained on empirical data available from experiments and simulations. The consequences of both accuracy and/or inaccuracy of the ML predictions stand at the same level as the accuracy of the empirical data used currently. Unfortunately, given technological experimental limitations in the state of the art, it is difficult to ascertain current uncertainties. The accuracy implications, although incredibly important, are indeterminate at this moment, and hence outside of scope of this work. As new data become available, it is conceivable that accuracy improvements would also materialize. The techniques demonstrated are limited by the same level of uncertainty associated with the empirical data. The community of practice has acknowledged the indeterminate quality of such data; however, future technical improvements in data collection would allow validation of both current uncertainties as well as accuracy. Similarly, once this is determined, the ML uncertainties and accuracies would also manifest themselves since they represent analogous answers.

III. Used Machine Learning Topics

A. Multivariate Gaussian

Probability distributions are useful tools to describe the center, spread, and shape of data. The Gaussian distribution is a good fit for unimodal data that have a clear center and the characteristic symmetric bell shape. The single-variate Gaussian is defined by two parameters: a mean μ that describes the data center, and a variance σ^2 that describes the data spread. It is defined as

$$\mathcal{N}(x|\mu,\sigma^2) = \frac{\exp(-(1/2)(x-\mu)^2/\sigma^2)}{\sqrt{2\pi\sigma^2}} \tag{1}$$

Higher moments for the single-variate Gaussian are constant, i.e., a skewness of zero and a kurtosis of three. Equation (1) can be extended to the multivariate case where data are in the form of vectors $\boldsymbol{x}$, requiring a mean vector $\boldsymbol{\mu}$ and a symmetric covariance matrix Σ. The multivariate Gaussian is defined as

$$\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu},\Sigma) = \frac{\exp(-(1/2)(\boldsymbol{x}-\boldsymbol{\mu})^T\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu}))}{\sqrt{(2\pi)^d|\Sigma|}} \tag{2}$$

where d is the dimension of the data [19].

Often, the distribution of a given dataset has a shape too complex for a single Gaussian (e.g., multimodal distributions). This effect is apparent when, for a given simulation case, the intersection point between fragment tracks and a spherical surface of defined radius R are plotted on a polar–azimuth map. The polar and azimuth angles ϕ and θ, shown on the "fragment location" inset in Fig. 3, can be calculated from a surface intersection point $[x, y, z]$ as

$$\phi = \tan^{-1}\left(\frac{z}{\sqrt{x^2+y^2}}\right)$$

$$\theta = \tan^{-1}\left(\frac{y}{x}\right)$$

As an example, a polar–azimuth map of track intersections through a sphere with a radius of 76 m from the simulation case with a detonation attitude of 90 deg in pitch, 0 deg roll and yaw, and a speed of 0 m/s is shown in Fig. 4, as well as the simulated fragment tracks and sphere intersection points scaled to a unit sphere.

B. Gaussian Mixture Models

Although the use of only one multivariate Gaussian would not make a good fit to the distribution in Fig. 4, it is possible to linearly combine several to yield a probability distribution that fits these more complicated multimodal data shapes. By performing a linear,

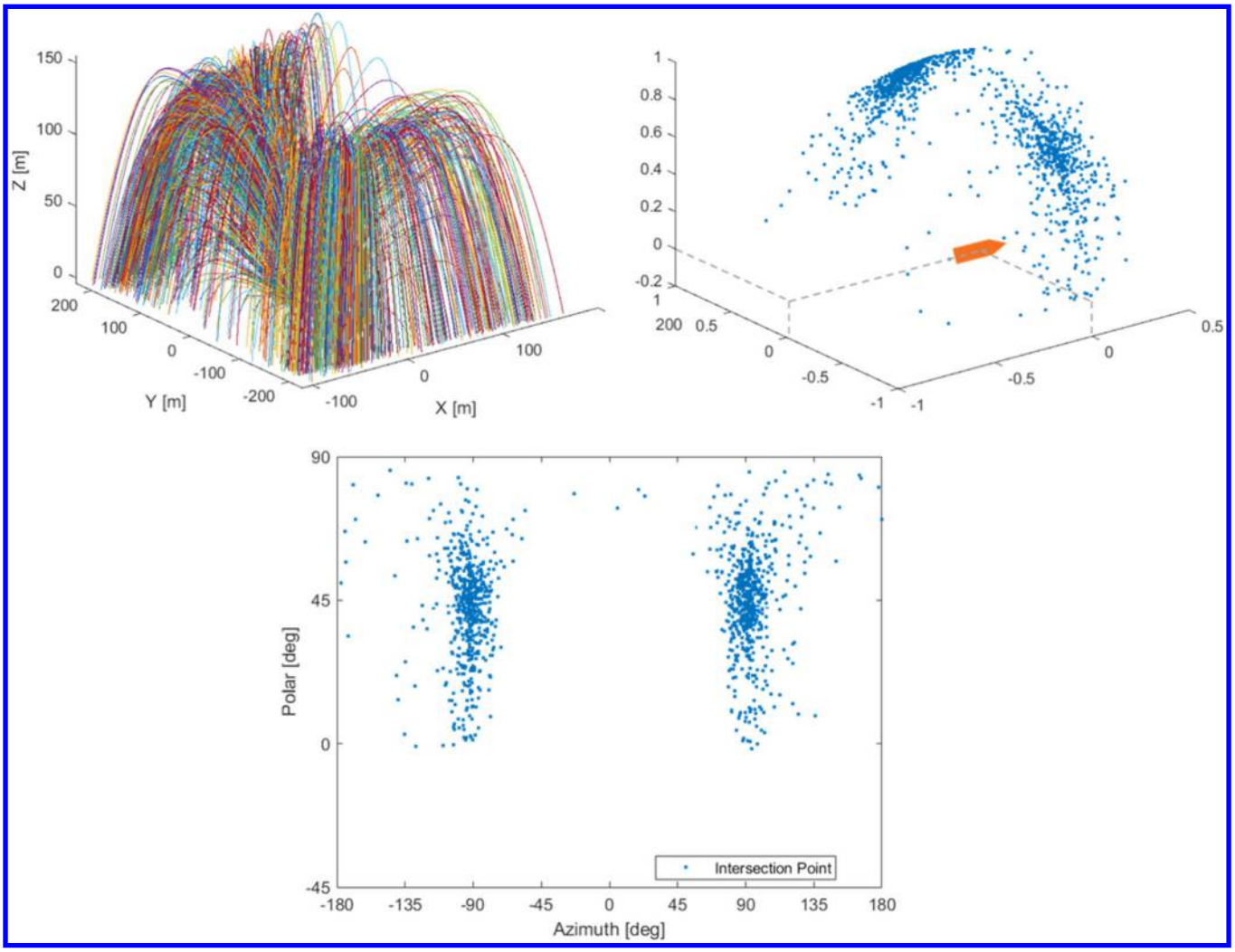

Fig. 4 Simulated fragment tracks after detonation (top-left), fragment track intersection points scaled to unit sphere (top-right), polar-azimuth map of simulation track intersection points (bottom).

weighted superposition of several multivariate Gaussian distributions, the Gaussian mixture model, which itself is a probability distribution, can be defined as

$$p(\boldsymbol{x}) = \sum_{k=1}^{N} \pi_k \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \Sigma_k)$$

where $\mathcal{N}$ is the multivariate Gaussian distribution from Eq. (2), N is the number of components used, and π_k are the mixing coefficients where $\sum_{k=1}^{N} \pi_k = 1$. The parameters that define a GMM therefore include N mixing coefficients π_k, N mean vectors $\boldsymbol{\mu}_k$, and N covariance matrices Σ_k. GMMs can fit a distribution of arbitrary shape for an infinite number of mixtures [19].

Directly fitting a probability distribution to a dataset is the process of determining the parameters that maximize the likelihood that the distribution predicts points in the dataset. Although there exists an analytical method to fit a multivariate Gaussian to a dataset, fitting a GMM to a dataset requires an iterative, numerical technique. The most commonly used technique (and the one used in this study) is Expectation Maximization (EM). Details on this technique can be found in the work of Bishop [19].

C. GMM Considerations

GMM hyperparameters are different from GMM parameters, which are the mixing coefficients, means, and covariances that define one specific GMM. The hyperparameters to choose when fitting a GMM to a dataset include the number of components N and the covariance type (i.e., full or diagonal). In the case of two-dimensional data, a GMM with N components is defined by a total of $6N$ parameters when full covariance matrices are used or $5N$ parameters when diagonal covariance matrices are used. Although the use of full covariance matrices requires an additional N parameters over the use of diagonal covariances, it allows the Gaussian distributions to fit well to distributions that are misaligned with the coordinate axes of the data. As an example, GMMs with four components, eight components, full covariance matrices, and diagonal covariance matrices were fit to a track intersection distribution; and the probability density function (PDF) contour plots are shown in Figs. 5 and 6. The example shown is for a simulated detonation with a pitch angle of 30 deg, yaw and roll angles of 0 deg, and a speed of 0 m/s.

D. Random Forest Regression

Random forest regressors are a useful regression tool that can be trained to map an input space to an output space. As computing power has progressed, random forests have been an attractive prediction tool where large datasets are available. Random forests are a machine learning technique that makes use of many decision trees to make predictions. Although a single decision tree f_i overfits its training dataset, random forests can avoid overfitting by fitting several decision trees on randomly selected training data points through the process of bagging. For a given input, a trained random forest g performs regression by taking the mean of all its individual trees

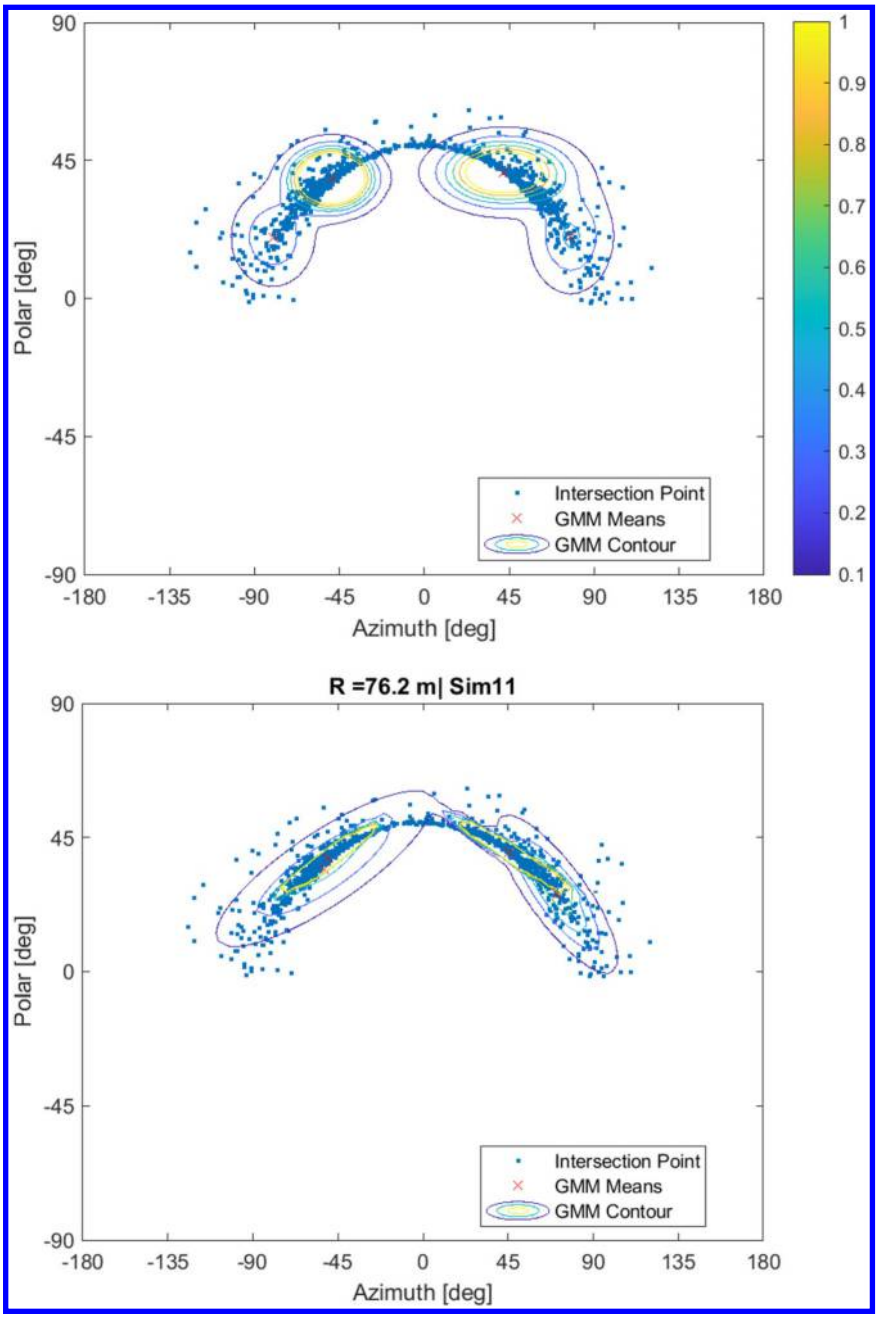

Fig. 5 Four-component GMMs fit to track intersections for both diagonal (top) and full (bottom) covariance matrices. Contour lines represent lines of constant probability density for GMM. Simulation case is for a static detonation at 30 deg pitch and 0 deg yaw and roll.

$$g(\boldsymbol{u}) = \frac{1}{M}\sum_{i=1}^{M} f_i(\boldsymbol{u})$$

where $\boldsymbol{u}$ is an input vector to the random forest, and the hyperparameter M is the number of trees used to build the random forest [20].

IV. Model Overview

A. Problem Statement

The goal of this research is to create a transfer function that can predict in-flight fragmentation behavior using static arena experimental data and high-fidelity numerical simulation data. Specifically, the tool will predict, based on a terminal state of the warhead, the number of fragments that pass through a defined surface. System terminal state information includes the warhead detonation attitude (roll, pitch, and yaw angles) and velocity. Surfaces are defined by a polar and azimuth range on the surface of a sphere of defined radius. Predictions are based only on the system terminal state for the specific case of 5 mm ball bearings. In this investigation, the methodology of predicting fragment counts is discussed in detail, and a version of the discussed methods can be explored in future investigations regarding mass and velocity predictions.

B. Architecture Overview

The proposed implementation to fulfill the goals set in the problem statement involves the use of a machine learning regressor (i.e., a random forest) to predict probability distributions of fragment track intersections on polar–azimuth maps. Given an input system terminal state and sphere radius, a random forest is used to predict the GMM parameters of the track intersection probability distribution and the total number of fragments. In other words, the random forest input $\boldsymbol{u}$ includes warhead terminal state information (i.e., roll angle, pitch angle, yaw angle, and speed) and a radius at which to predict a fragment track intersection distribution. The random forest output $g(\boldsymbol{u})$ contains the GMM means $\boldsymbol{\mu}_k$, covariances Σ_k, mixing coefficients π_k, and the total number of fragments reaching and passing

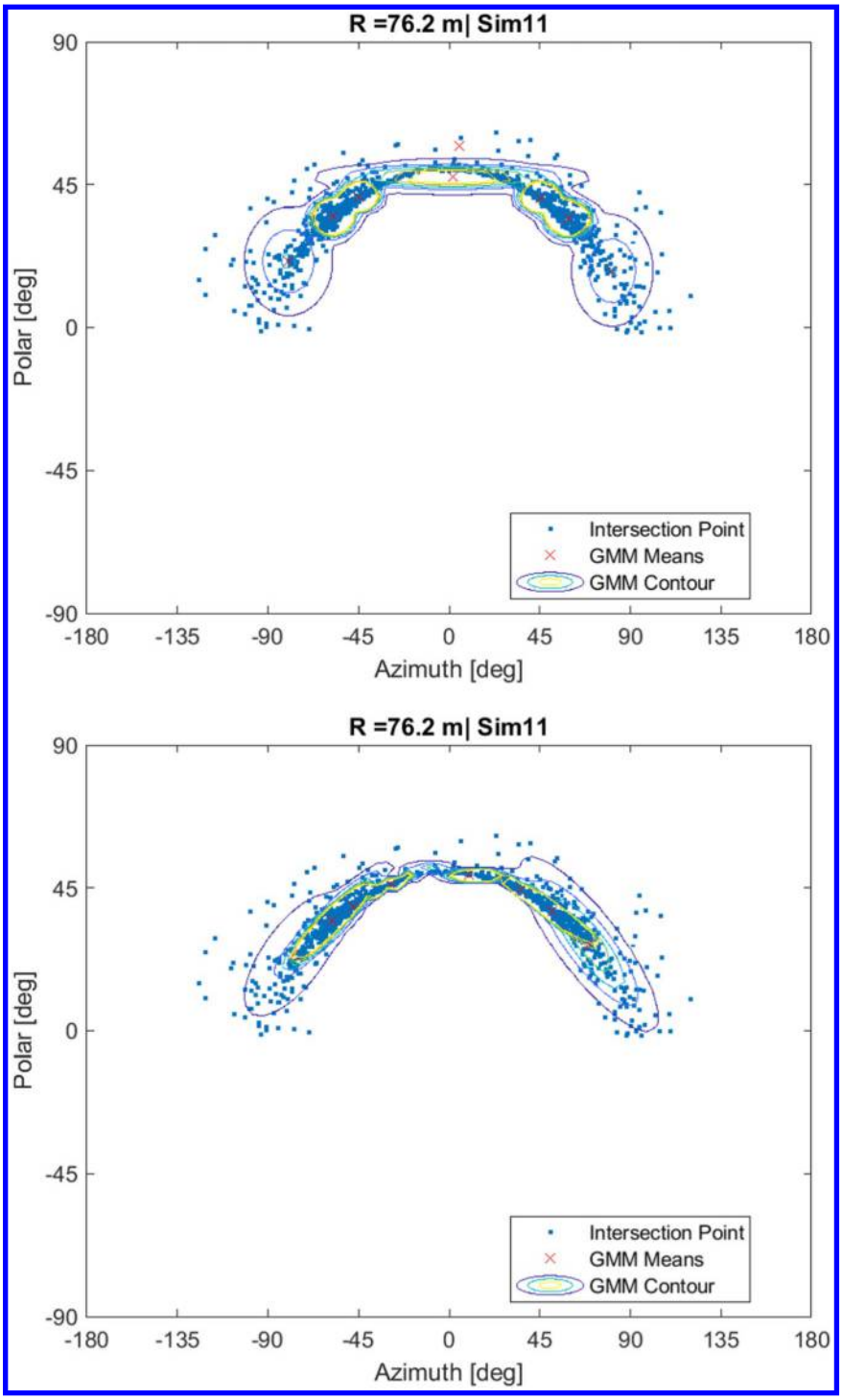

Fig. 6 Eight-component GMMs fit to track intersections for both diagonal (top) and full (bottom) covariance matrices. Contour lines represent lines of constant probability density for GMM. Simulation case is for a static detonation at 30 deg pitch and 0 deg yaw and roll.

through the sphere's entire surface N_{total}. The weapon type is not currently used as a training input since simulation data are only available for one simulated warhead with ball-bearing fragments.

Using this predicted GMM and a given surface defined by polar–azimuth boundaries, an integration of the GMM can be performed to calculate a predicted number of fragments crossing this region of the sphere surface. For a given probability distribution $p(\boldsymbol{x})$ of fragment track intersections on the entire surface of a sphere, and given a region S on the sphere surface, the number of fragments that pass through the defined surface region is

$$N_{\text{frag}} = N_{\text{total}} * \int\int_S p(\boldsymbol{x})\mathrm{d}S$$

where N_{total} is the total number of fragments passing through the sphere's entire surface; and $dS = d\phi d\theta$, where θ is the azimuthal angle and ϕ is the polar angle. The total volume under the multivariate probability distribution represents all track intersections, whereas the volume under a specific region represents the proportion of fragments that cross in that region. A diagram of the implemented model is shown in Fig. 7.

C. Generating Training Data

Regressor training data are generated from 1100 simulation results, based on NAWCWD research software. For each simulation, a set of radii are generated. Looping through each radius, polar–azimuth maps of track intersections are generated, GMMs

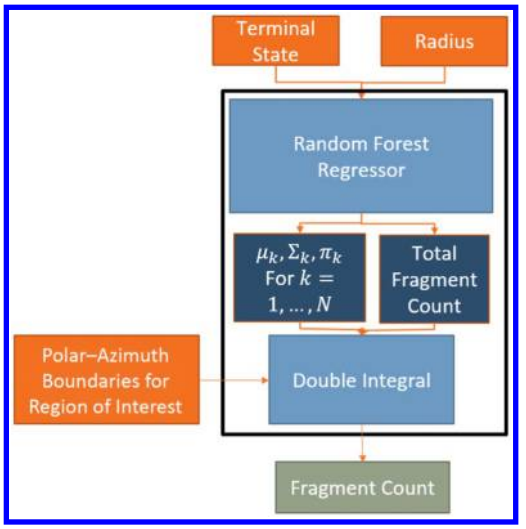

Fig. 7 Diagram of proposed model to predict fragment counts.

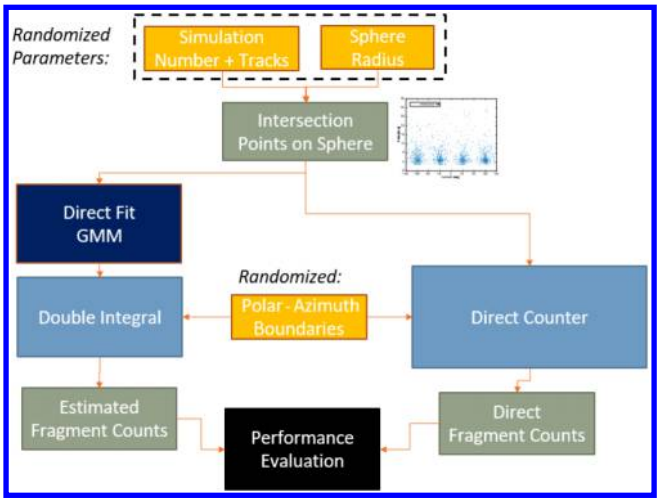

Fig. 8 Diagram of Monte Carlo method evaluating GMMs fit directly to track intersection data.

are fit to the intersection distributions via expectation maximization [19,21], and the total number of intersections is counted. The system terminal state (i.e., attitude and velocity) associated with the simulation case and the randomly generated radius are saved as regressor training inputs to the regressor. The GMM parameters and the total intersection count are saved as regressor training outputs. The "test data" for the ML model are extracted from randomly selected track intersection distributions not included in the training dataset, as is discussed with the performance evaluation methods (Sec. V).

It should be noted that the EM is highly dependent on its initialization since it finds a local optimum (not the global optimum). By default, many EM implementations randomly initialize the mean vector locations. With randomized initialization, the *i*th mean vector may not converge to the same location for repeated fits to the same dataset. Additionally, mean vectors may trade their convergence points for repeated fits to the same dataset. This lack of consistent ordering poses a problem for training any regressor to predict GMM parameters. The random forest would not properly learn which components correspond to which outputs. It is therefore necessary to implement a strict initialization strategy for fitting GMMs during training data generation. In this study, GMM components were initialized at a specified point on the sphere surface so that they converge deterministically in a specific order. This initialization provides consistency in the ordering of the GMM components predicted by the random forest.

V. Performance Evaluation Methods

A. Direct Fit GMMs

The first set of tests on the model is to strictly evaluate the performance of GMMs. It is imperative to understand how good of a fit GMMs are to the intersection distributions. This evaluation is done via a Monte Carlo simulation that compares fragment counts predicted by integration of a GMM fit directly to intersection distributions versus direct counts of the intersections.

First, fragment tracks are pulled from a random simulation case, and a single radius and polar–azimuth range are randomly generated. A GMM is then directly fit via expectation maximization to the track intersection distribution on the generated radius. The GMM is integrated on the generated polar–azimuth range and multiplied by the total number of directly counted intersections to get a predicted fragment count. The number of fragments that pass within the polar–azimuth range are also directly counted. The predicted counts are compared to the direct counts, and a 95% confidence error analysis is performed. Higher moments (i.e., skewness and kurtosis) are calculated, and the Anderson–Darling test for normality is performed [22]. A diagram of the implemented Monte Carlo method is shown in Fig. 8.

B. Full Model Monte Carlo Method

The capability of the chosen machine learning regressors to predict GMMs that accurately describe the track intersection distributions on a sphere surface for a given system terminal state is determined by evaluating the performance of the full model described via a Monte Carlo simulation.

First, the system terminal state is pulled from a random simulation, and a radius and polar–azimuth range are randomly generated. These randomly generated conditions correspond to track intersection distributions not seen in the random forest training dataset and can therefore serve as the test dataset. A direct count of fragments through the polar–azimuth range is calculated from the simulation data. Then, the trained regressor [23] uses the system terminal state and radius to predict the total fragment count and the GMM parameters. The GMM is integrated and multiplied by the predicted total fragment count to get a predicted count of fragments through the defined surface. The predicted counts are compared to the direct counts, and a 95% confidence error analysis is performed. Higher moments (i.e., skewness and kurtosis) are calculated, and the Anderson–Darling test for normality is performed [22]. After many runs with randomized conditions, the count error distribution is plotted as a PDF for one of the cases. A diagram of the Monte Carlo method described is shown in Fig. 9.

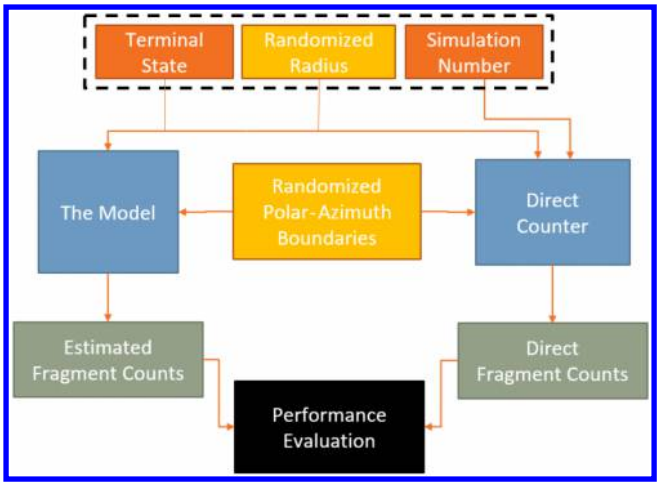

Fig. 9 Diagram of the Monte Carlo method to evaluate the performance of the full model.

VI. Monte Carlo Numerical Results

A. Direct Fit GMMs

The performances of GMMs are evaluated for four different sets of hyperparameters, shown in Table 2. One could use many more GMM components than were used in the four cases listed; however, using too many components in the GMM could overfit a specific track set and radius. Additionally, the random forest regressor in this model is tasked with predicting a higher-dimensional space than its input.

Although the four-component diagonal covariance GMMs have visibly poorer fits than the other three GMM types, Monte Carlo simulations are still run on cases 1 and 2 in order to evaluate the fit numerically and not just visually. Additionally, it may be the case that higher-order GMMs fit the data better, but it would be wrong to assume the random forest can predict low- and high-order GMMs equally well. This does, however, turn out to be true for the cases used. The 95% confidence interval, standard deviation, skewness, and kurtosis for the fragment count errors from the direct fit Monte Carlo numerical simulations are provided in Table 3. Generally speaking, good performance of the model is characterized by a confidence interval that includes zero error and a small standard deviation.

Performance of the GMM increases when the number of components is increased from four to eight, as seen in the results in Table 3. The confidence interval includes zero error for the two diagonal covariance cases (cases 1 and 3); however, there is a tradeoff with the lower standard deviations in the full covariance cases (cases 2 and 4). The substantial decrease in the error standard deviation indicates that fewer of the Monte Carlo runs resulted in large count errors. Additionally, the Anderson–Darling normality test results in rejection of the hypothesis that the error distribution is Gaussian for all four cases. Although the difference in standard deviations resulting from switching the covariance type from diagonal to full is smaller than that which results from increasing the number of components, it is enough to justify running the full model Monte Carlo simulation on all four cases.

B. Full Model Monte Carlo Method

Although the results of the direct fit GMM Monte Carlo simulations represent how accurate of a fit GMMs are to the intersection distributions, the ability of the random forest regressor to predict the GMMs is evaluated via the Monte Carlo method on the full model. The 95% confidence interval, standard deviation, skewness, and kurtosis for the fragment count errors from the full model Monte Carlo numerical simulations are provided in Table 4. A PDF of the count error distribution that results from the full model Monte Carlo simulations performed is provided for case 4 in Fig. 10.

As expected, the performance of the predicted GMM is not as accurate as the directly fit GMM. This reduction in performance is indicated by the increase in the standard deviation of count differences. One similarity between the direct fit Monte Carlo simulation and the full model Monte Carlo simulation is that case 4 (i.e. eight components and full covariance) yields the lowest standard deviation of fragment count differences. Only case 1 yields a confidence interval that includes zero error; however, there is a tradeoff with it having the highest standard deviation of count errors. Additionally, the Anderson–Darling normality test results in rejection of the hypothesis that the error distribution is Gaussian for all four cases.

To better understand which Monte Carlo simulation runs yield high count differences, an example of a high count difference can be examined. Plots of the intersection distribution, predicted GMM, and directly fit GMM are shown in Fig. 11. For a given polar–azimuth range of this specific run, the model predicted 222 fragments, whereas 80 are counted directly (difference of −130). The predicted total fragment count is 1985, whereas the directly counted total is 1997 (difference of only 12).

Although visual inspection of the regressor predicted GMM might indicate it is a good fit to the intersection distribution, the Monte Carlo simulation run allowed for the realization of high count differences. At close inspection of the plots in Fig. 11, the contours of the GMMs can be seen. The direct fit GMM shows high-valued contours passing through a smaller portion of the polar–azimuth boundary rectangle than the high-valued contours of the regressor predicted GMM. Higher-valued contours passing through a larger portion of the integration box yield a higher fragment count predicted by integration of the GMM. Even when the predicted GMM is a close match to a GMM directly fit to the intersection distribution, slight differences in the contours can lead to a few high count differences in polar–azimuth boundaries on the edge of the distribution. Despite these few instances of high count differences, the random forest regressor has demonstrated its ability to learn to predict the parameters of a GMM that accurately describes the fragment distribution for a given warhead system terminal state.

Since the intended application of the developed technique is to allow for fast prediction of fragment flyouts, some discussion and comparison of the computational runtimes is warranted. It should be noted that the machine used to run the detonation simulations is

Table 2 Hyperparameter variation for GMMs

Case	Number of components	Covariance type
1	4	Diagonal
2	4	Full
3	8	Diagonal
4	8	Full

Table 3 Error statistics from the GMM direct fit Monte Carlo numerical evaluations

Case	95% confidence interval	Standard deviation	Skewness	Kurtosis
1	[−0.18, 0.06]	20.0	1.4	30.0
2	[−0.31, −0.12]	16.0	0.97	29.0
3	[−0.046, 0.071]	9.9	1.5	41.0
4	[−0.093, −0.0022]	7.7	0.41	26.0

Table 4 Mean and standard deviation of fragment count differences from full model Monte Carlos

Case	95% confidence interval	Standard deviation	Skewness	Kurtosis
1	[−0.75, 0.15]	32.0	−1.0	34.0
2	[−0.86, −0.029]	30.0	−1.5	35.0
3	[−1.0, −0.36]	23.0	−2.0	49.0
4	[−1.1, −0.43]	22.0	−2.1	59.0

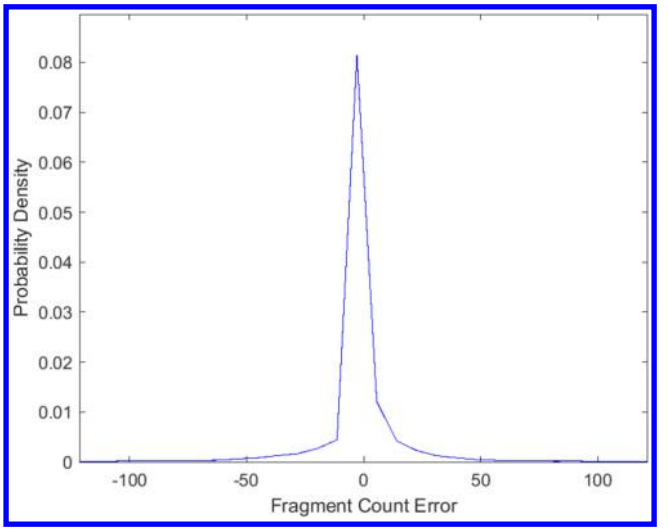

Fig. 10 Histograms of count differences from full model Monte Carlos for case 4.

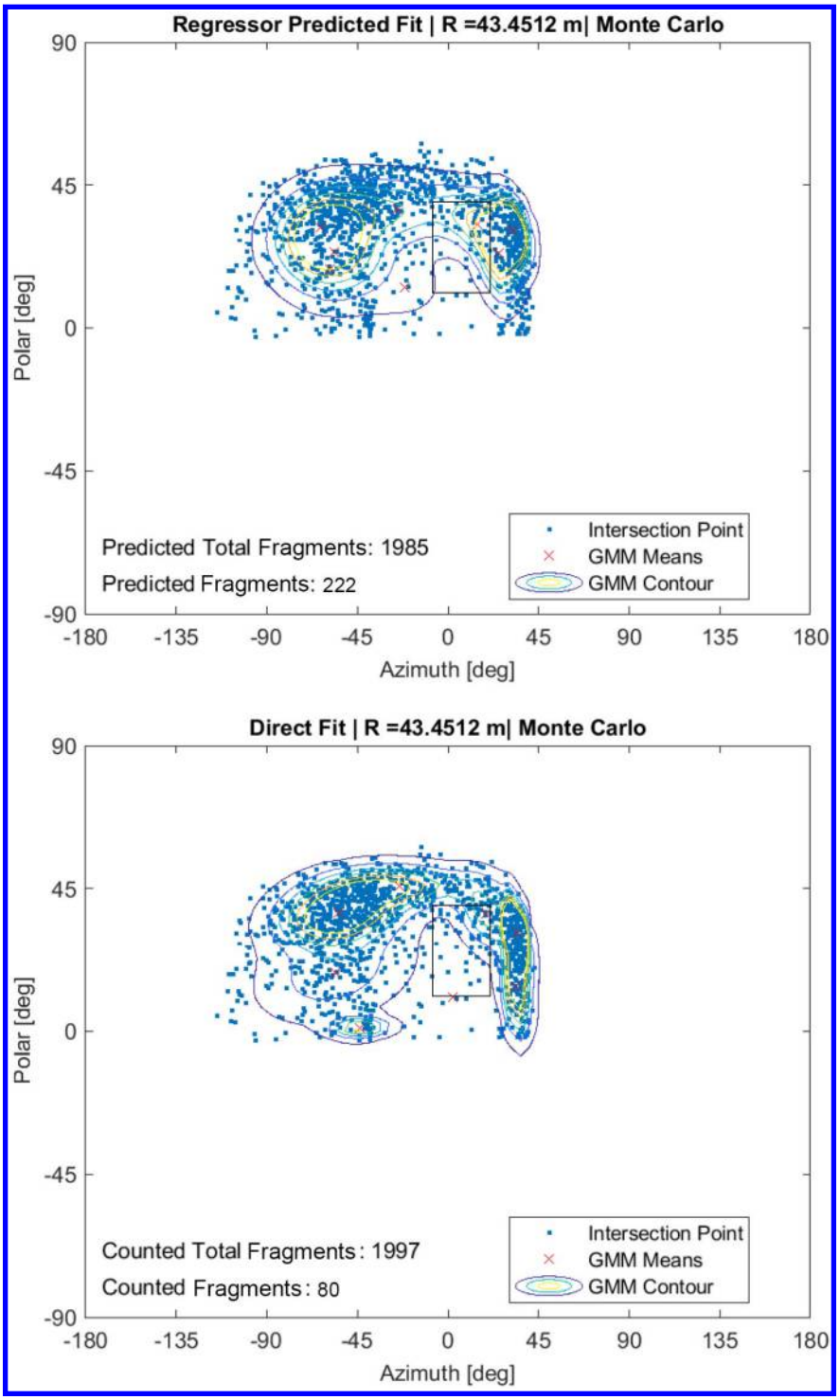

Fig. 11 Monte Carlo simulation run resulting in high count difference. The black rectangle is the polar–azimuth boundary for this case.

different from the machine used to train and run the model developed in this study. The simulation setup time is measured in minutes, whereas runtimes can vary; depending on the scenario being computed, the runtime ranged from minutes to up to 2 h. The machine learning computational time is similarly split into two constituents: in this case, the training and prediction runtimes. Setup and training of the machine learning algorithm ranged from 1 to 3 min, whereas the trained model takes less than 1 s to predict a fragment intersection distribution at a single radius. It should be stressed that this computation time comparison is only qualitative since 1) the two techniques are not exact analogies because they focus on different computational regimes, 2) the developed model is not intended to be a replacement for the high-fidelity simulations given their different applications, and 3) both the high-fidelity simulations and machine learning algorithms are still developmental in nature, and hence there is opportunity for code optimization in the future.

VII. Assimilation with Experimental Data

So far, the model performance has only been assessed in the prediction of fragment–surface intersection distributions appertaining to simulation data. Although the Monte Carlo simulations used to evaluate the model performance showed that the random forest regressor is able to predict distributions that match what is seen in simulation data, it is important to recall the ultimate goal of predicting

real-world behavior of warhead fragment flyout. Although the data from dynamic warhead detonations are not available, it is still possible to resolve the difference between the model predictions and the static arena experimental data, and then make qualitative assessments of predicted fragment–surface intersection distributions in various high-speed cases. An assessment of the distribution features in the simulation and experimental data can be used to determine how this assimilation must be performed. Another Monte Carlo numerical simulation can be used to evaluate the validity of the assimilated model.

A. Distribution Features: Simulation Versus Experimental

Although simulation cases are available for various detonation attitudes and speeds, the real 5 mm ball-bearing experimental data are available for the specific case of pitch, yaw, and roll angles of 0 deg, as well as a speed of 0 m/s. The track intersections distribution from the simulation for this specific static case as well as the track intersection distribution from the experimental data are shown in Fig. 12. The distribution is shown on both the surface of a unit sphere and on a polar–azimuth map.

There are a few key feature differences that can be observed in the intersection distributions. The features that both distributions share are the columns of intersection points at azimuth angles near ± 90 deg. One example of a feature of the experimental data distribution tha tis not present in the simulation data distribution is the presence of intersections on the $+X$ side of the detonation. On a polar–azimuth map, this feature corresponds to intersection points seen at azimuth angles close to zero. One example of a feature present in simulation data distributions not seen in the experimental data distribution are the sections of intersection columns at polar angles above 60 deg and below about 5 deg. Missing regions in the experimental data distributions are expected since data are only available for some of the fragments that pass through the camera fields of view. However, the regions of data not present simulation data distributions

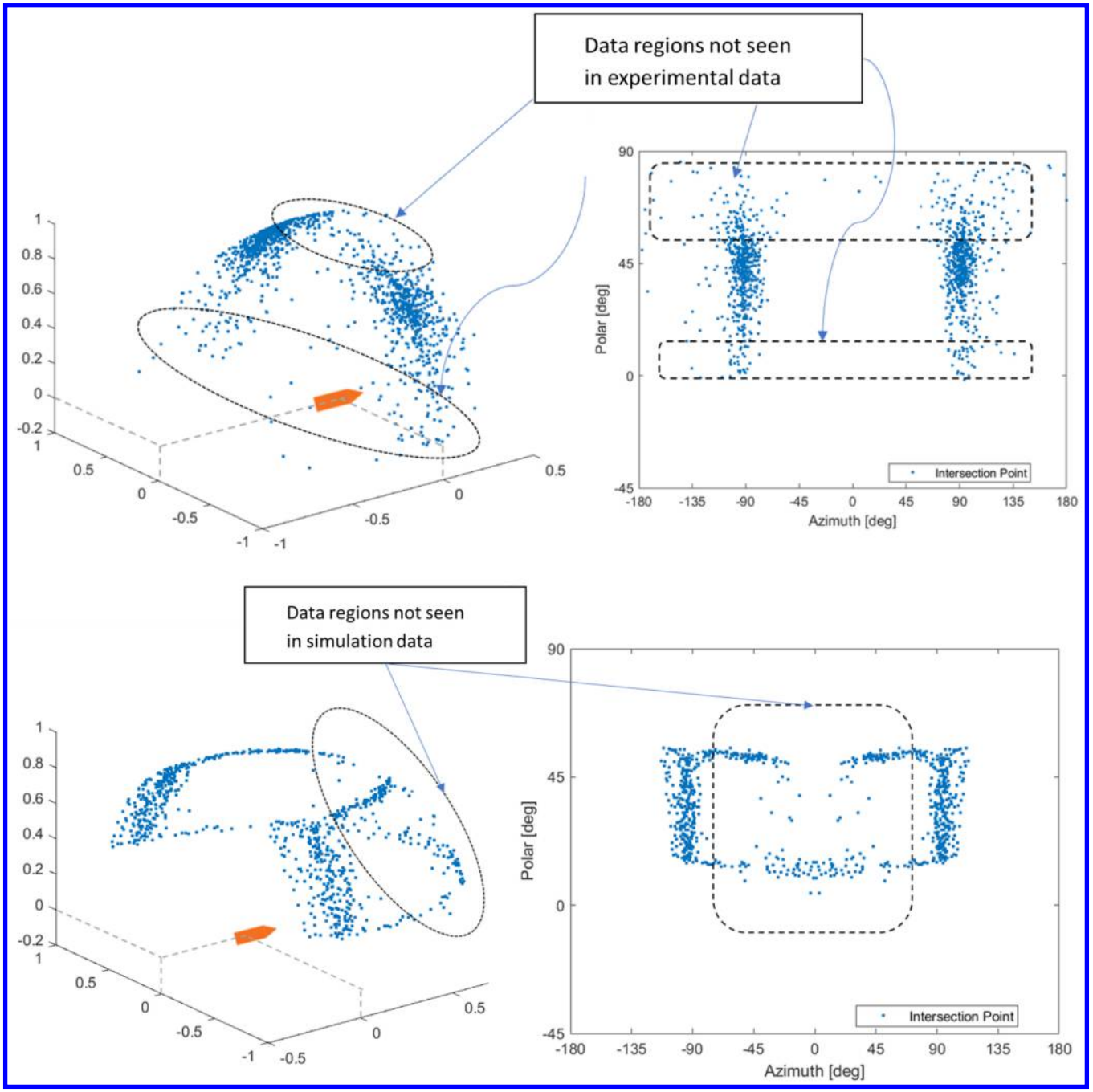

Fig. 12 Track intersection distributions shown on polar–azimuth map and scaled to unit sphere for static simulation case (top) and real static test (bottom).

directly affect model predictions when included in the training set for the random forest. Therefore, a method of assimilating the simulation and experimental data is needed so that the regressor can predict features of both.

B. Assimilation Method and Assessment

The discrepancy in the distribution features seen in simulation and experimental data can be resolved by assimilating random forest training data. GMMs fit to experimental data are combined into the dataset of GMMs fit to simulation data. A Monte Carlo method that randomizes the polar–azimuth boundary of integration on a single GMM prediction is used to assess the performance of the assimilated model. Each sample from experimental data corresponds to a radius at which regressor training data are produced, and samples not in the training set are used as a test set in the Monte Carlo simulation.

Assessment is done by comparing count proportions and visual inspection for specific distribution features. In the previously presented Monte Carlo assessments, model predicted counts are compared to direct counts from simulation data for performance evaluation. This evaluation technique is not appropriate for predictions compared to experimental data because the total number of fragment tracks in experimental data is close to 400 (close to 800 when duplicated), and 3681 fragments are simulated. Since most of the random forest's training set comes from simulation data, the regressor always predicts higher fragment counts than what appears in the experimental dataset. The "integration error" is used in place of the count difference, and it is defined as

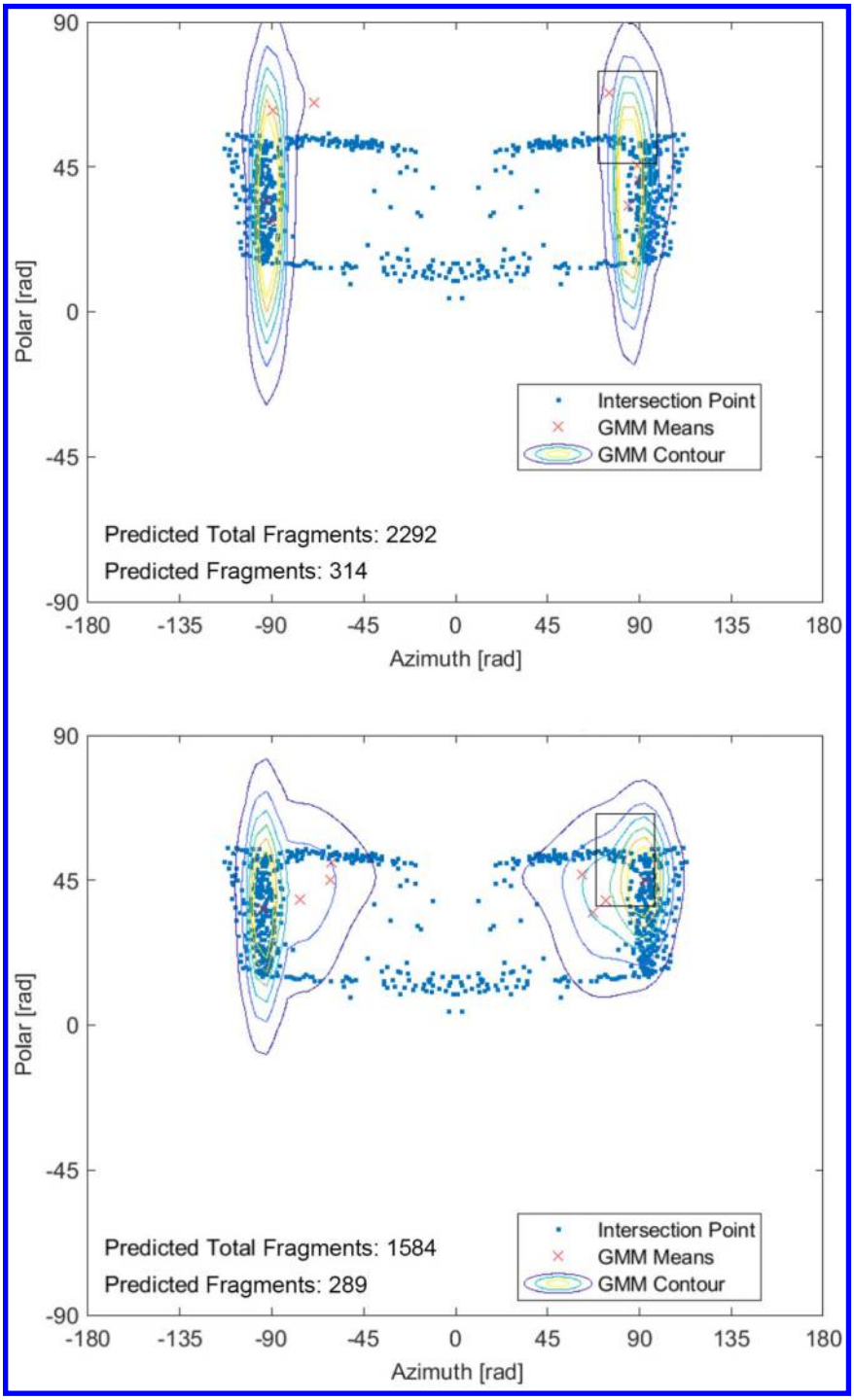

Fig. 13 Predicted GMMs from random forests trained with (from top to bottom) 0, 100, 200, and 400 experiment samples in its training set. GMM contours are overlaid onto experiment intersection data.

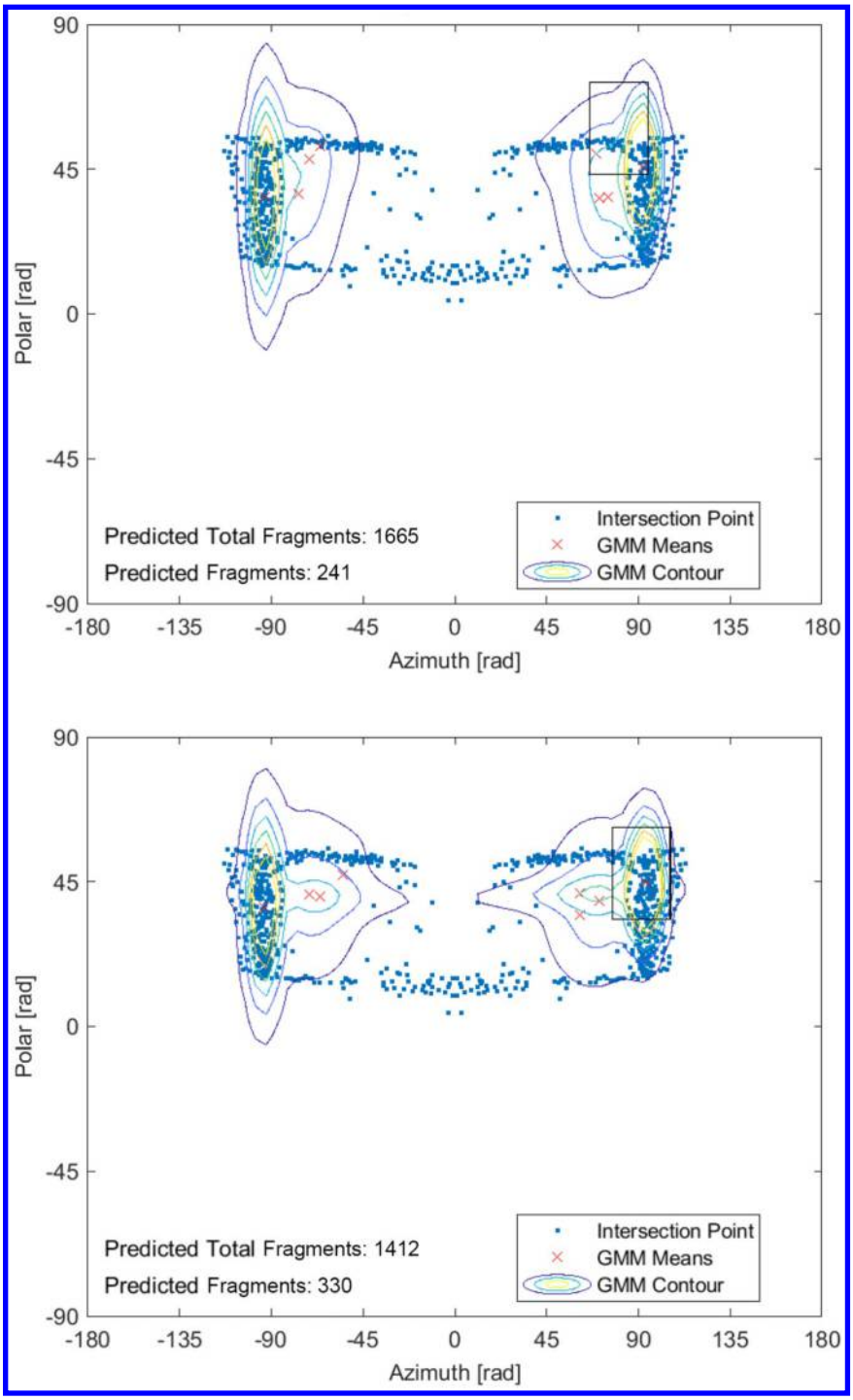

Fig. 13 (Continued)

$$I_{\text{error}} = \frac{N_{\text{direct}}}{N_{\text{direct,total}}} - \int\int_S p(\boldsymbol{x})\mathrm{d}S \tag{3}$$

where p is the GMM predicted by the random forest. A 95% confidence interval is calculated for the integration error distribution. Higher moments (i.e., skewness and kurtosis) are calculated, and the Anderson–Darling test for normality is performed [22].

The tradeoffs of this assimilation method and the analysis of an assimilated model's performance should be noted. It is not time consuming to retrain a random forest compared to the time required to preprocess simulation data into training sets. Also, allowing the random forest to learn directly from experimental data guarantees that it will learn some features from the static arena detonations. Including experimental data in the training set for the random forest, however, could lead to overfitting. It is possible that the random forest may simply learn the features of the test data and may not learn the general patterns of the distributions. The best validation for the assimilated model will come from a real dynamic detonation test. At present, evaluation can be performed from the prediction of static cases and the visual inspection of features in dynamic cases.

C. Assimilation Results and Predictions

The number of samples from experimental data corresponds to the number of different radii at which track intersection distributions were used to fit GMMs. The Monte Carlo method evaluating the GMM prediction from a random forest trained with zero experimental samples was performed first to be used as a reference. Then, the Monte Carlo method was run for various amounts of experimental

Table 5 Error analysis for integration errors evaluated from Eq. (3)[a]

Experimental samples assimilated into training set	Confidence interval	Standard deviation	Skewness	Kurtosis
0	[−0.0072, 0.37]	3.1	−0.01	4.8
100	[0.37, 0.64]	2.2	0.64	5.3
200	[0.34, 0.65]	2.5	0.53	4.8
400	[0.37, 0.64]	2.1	0.88	5.1

[a]The confidence intervals and standard deviations are given as percentages.

data samples included in the random forest training set, from 75 to 400. The predicted GMM for the static case with 0 deg roll, pitch, and yaw at a 7.62 m radius is shown as a contour plot overlaid onto experimental sphere intersection points. The plots for the cases of 0, 100, 200, and 400 experiment samples included in the training set are shown in Fig. 13.

In the case of only zero experimental samples included in the training set, the predicted GMM closely matches the distribution from the simulation data seen in Fig. 12, and there are no distinct features of the experiment distribution visible, as expected. After increasing the number of included samples to 100, the predicted GMM shape begins to morph away from the distribution shape seen in just the simulation data. After increasing the number of included samples to 200, the predicted GMM shape shows features of both the simulation and the experiment data. The 95% confidence interval, standard deviation, skewness, and kurtosis for the fragment integration error from Eq. (3) for each Monte Carlo simulation are shown in Table 5. Again, count differences would not be appropriate for the assimilated model since the random forest is expected to greatly overpredict the number of fragments seen in experimental data because not all fragments are tracked in experimental data.

The Anderson-Darling normality test results in rejection of the hypothesis that the integration error distribution is Gaussian for all cases. Since performance of the assimilated model is roughly characterized by a mean integration error close to 0% and small standard deviations, Table 5 shows that as more experiment samples were added to the training set, performance of the model increases up to a point between 100 and 400 added experimental samples. This characterization of good performance should be used cautiously, however, to avoid the effects of overfitting. A good model should not be expected to predict the experiment data perfectly, since the experiment data itself is not perfect, and it has regions of missing data where the cameras are unable to track fragments. However, combining performance indicated by the integration difference with the inspected features in the predicted GMMs can indicate that the random forest is predicting good GMMs without overfitting its training set.

In addition to the evaluation of the features present in the predicted GMM, an assessment of the symmetry assumed on the experiment data, and the preservation of this symmetry in the predicted GMMs is useful. The GMM predicted from a random forest trained on both simulation and experiment data can be integrated across a hemisphere. An integral yielding a value of 0.5 indicates total preservation of symmetry. A plot of the predicted GMM with a box around the region integrated is shown in Fig. 14.

The integral yields

$$\int_0^{\pi}\int_{-\pi/2}^{\pi/2} p(\boldsymbol{x})\,\mathrm{d}\phi\mathrm{d}\theta = 0.5054$$

where ϕ is the polar angle, and θ is the azimuthal angle. This integration indicates that a small amount of symmetry is lost during the assimilation of the data. However, this small amount of lost symmetry means that, in general, the random forest is able to preserve symmetry in its predictions.

The ultimate goal of this investigation is to create a prediction tool for the characterization of warheads detonating at high speeds. For

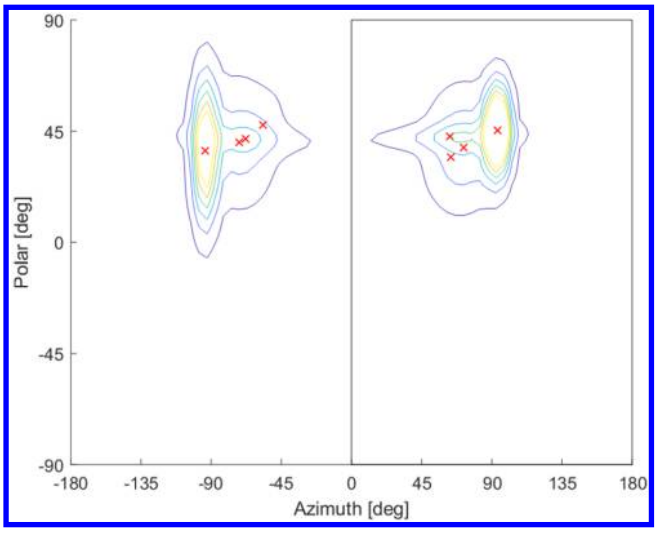

Fig. 14 Predicted GMM with assimilated data integrated in the shown box to evaluate preservation of symmetry.

the case of the same attitude of 0 deg pitch, yaw, and roll, a GMM can be predicted by the random forest for various speeds (here presented between 0 and 1524 m/s at 305 m/s intervals). Plots of the predicted GMMs for a radius of 7.6 m and intersection points randomly generated from the predicted GMMs for detonations at these various speeds are shown in Fig. 15. The plots show that as the warhead speed parallel to the ground increases, the spread of the intersection distribution decreases and converges to the region corresponding to the direction of the warhead's speed.

The results of the presented attempt at assimilating experimental and simulation data into a single training set show that the random forest is capable of learning to predict GMMs with an appropriate combination of features from both datasets in the static case. In dynamic cases, the assimilated model predicts distributions that change in accordance with physical expectations. Namely, higher-speed scenarios yield spatial distributions in more dense clusters in the direction of motion than do lower-speed detonations. The novel method explored in this paper represents the transfer function that could potentially predict high-speed fragment flyout behavior, despite only static experimental detonation data being available and while leveraging high-fidelity numerical flyout calculations. The GMM prediction and integration to yield a fragment count represent a new ability to quickly estimate collateral damage and lethality for a given detonation scenario. Dynamic warhead detonation experimental data would be used to further provide confidence in verifying and potentially validating the model.

VIII. Future Work

Future work will be directed toward characterizing model performance on the prediction of data available from static arena tests that do not use ball bearings. Data from static arena tests with natural fragmentation will be incorporated into the model. In addition, a method to predict mass and velocity distributions of fragments crossing a sphere surface will be implemented via a modified version of the techniques described in this paper. Velocities of fragments that intersect the sphere surface will be considered weights on the intersection point, and a version of expectation maximization developed by Gebru et al. [24] for weighted data will be used to fit GMMs to the fragment intersection distributions. A similar technique can be used for prediction of mass distributions as well. Other future work can explore methods to incorporate fragmentation theory into the model. Another popular regressor that predicts mixture models is the mixture density network [25]. This neural network architecture can be explored in a future investigation. Another potential project is the exploration of mixture models that are made from probability distributions other than the multivariate Gaussian. Other distributions

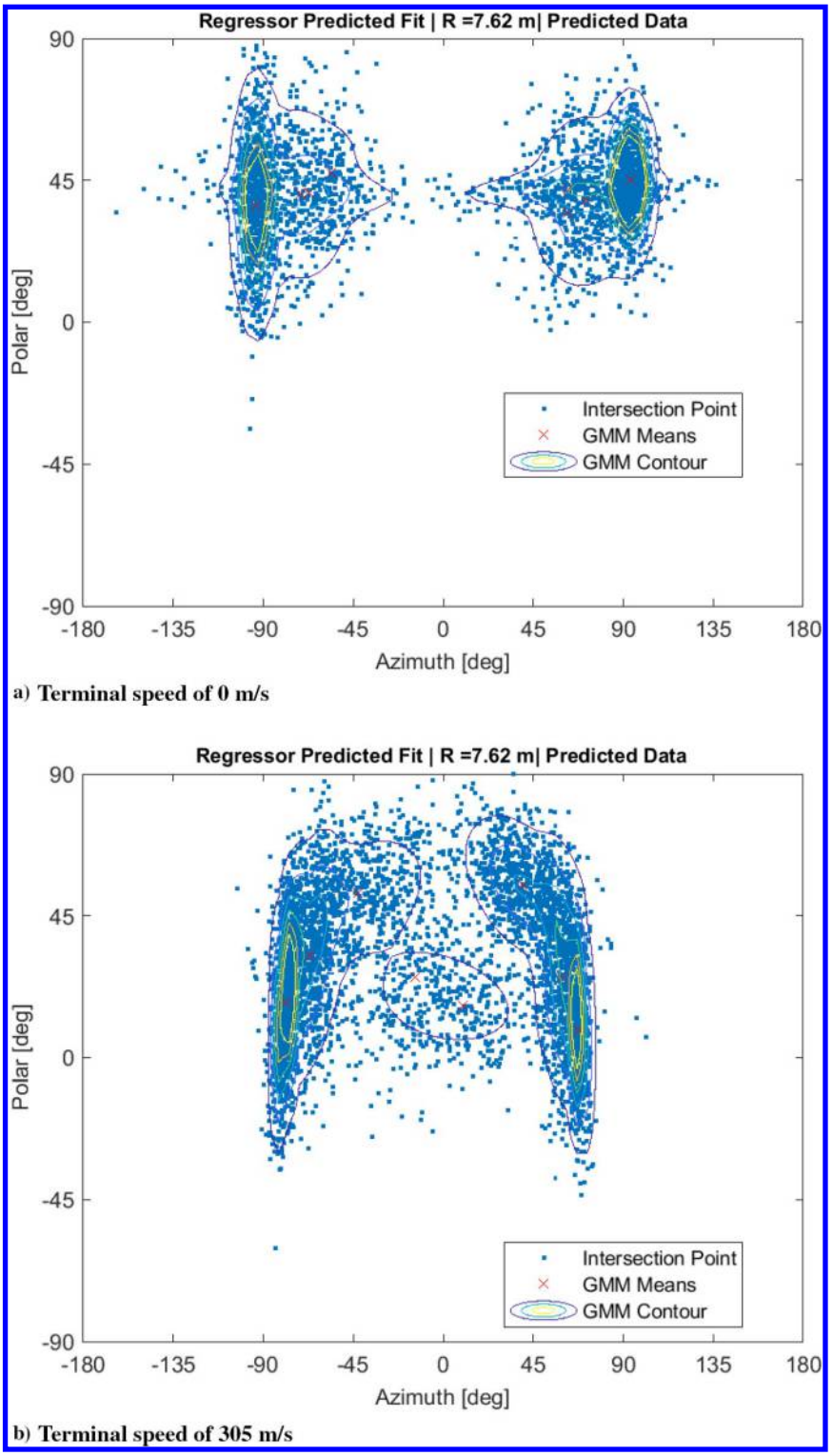

Fig. 15 Predicted distributions and randomly generated intersections for detonations at (from Figs. 15a to 15a) 0, 305, 610, 915, 1220, and 1525 m/s.

include the Poisson, gamma, and t distributions. A generalized mixture model can be defined as

$$p(\boldsymbol{x}) = \sum_{k=1}^{N} \pi_k P(\boldsymbol{x}|\boldsymbol{\Theta}_k)$$

where P is the base probability distribution that composes the mixture model, and $\boldsymbol{\Theta}_k$ are the distribution parameters (e.g., the mean and covariance for the Gaussian) [19].

The methods described predict the spatial distributions of fragments given a set of training data. Because the training data (which at present come primarily from dynamic simulations) as well as the methods used are refined, the fragment distribution prediction performance as evaluated on real experimental data will improve. These novel techniques will ultimately be verified once dynamic warhead detonation experimental data become available. As it stands, it was demonstrated that experimental and simulation datasets can be assimilated into a model that can perform fast prediction of fragment flyout from both static and high-speed terminal configurations. These fragment flyout predictions can be used for fast estimation

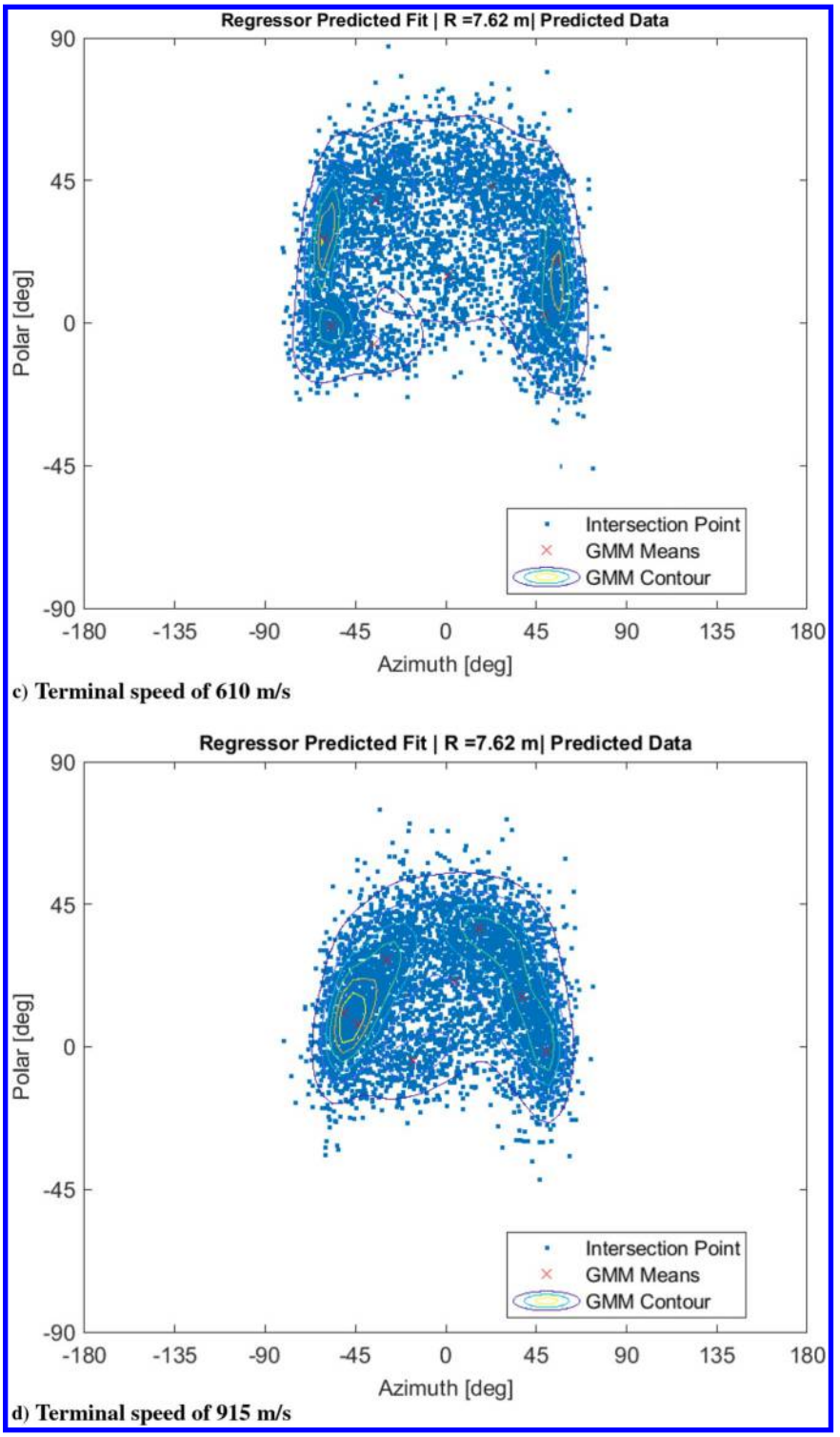

Fig. 15 (Continued).

of lethality and collateral damage caused by the high-speed detonation of warheads, allowing for more informed decision-making regarding warhead application to targets. Better informed decisions regarding warhead application will lead to reduced unintended damage to structures and personnel that surround specific targets in combat zones.

IX. Conclusions

This study aims to investigate techniques to predict warhead fragment flyout characteristics, making use of fragment tracks from dynamic simulation data and a real static detonation test. A random forest regressor is successfully trained to predict the number of fragments that passes through a sphere surface of a given radius as well as the probability distribution of fragment intersections on that sphere surface in the form of a Gaussian mixture model. The GMMs are integrated over some polar–azimuth range and multiplied by the predicted total number of fragments to yield a fragment count that passes through a specific region on the sphere surface. Monte Carlo simulations are used to evaluate the performance of the developed model, showing that it is capable of accurately predicting fragment intersection probability distributions based on its training set. Generally, an eight-component GMM with full covariances fits well to intersection data, and a trained random forest regressor is able to

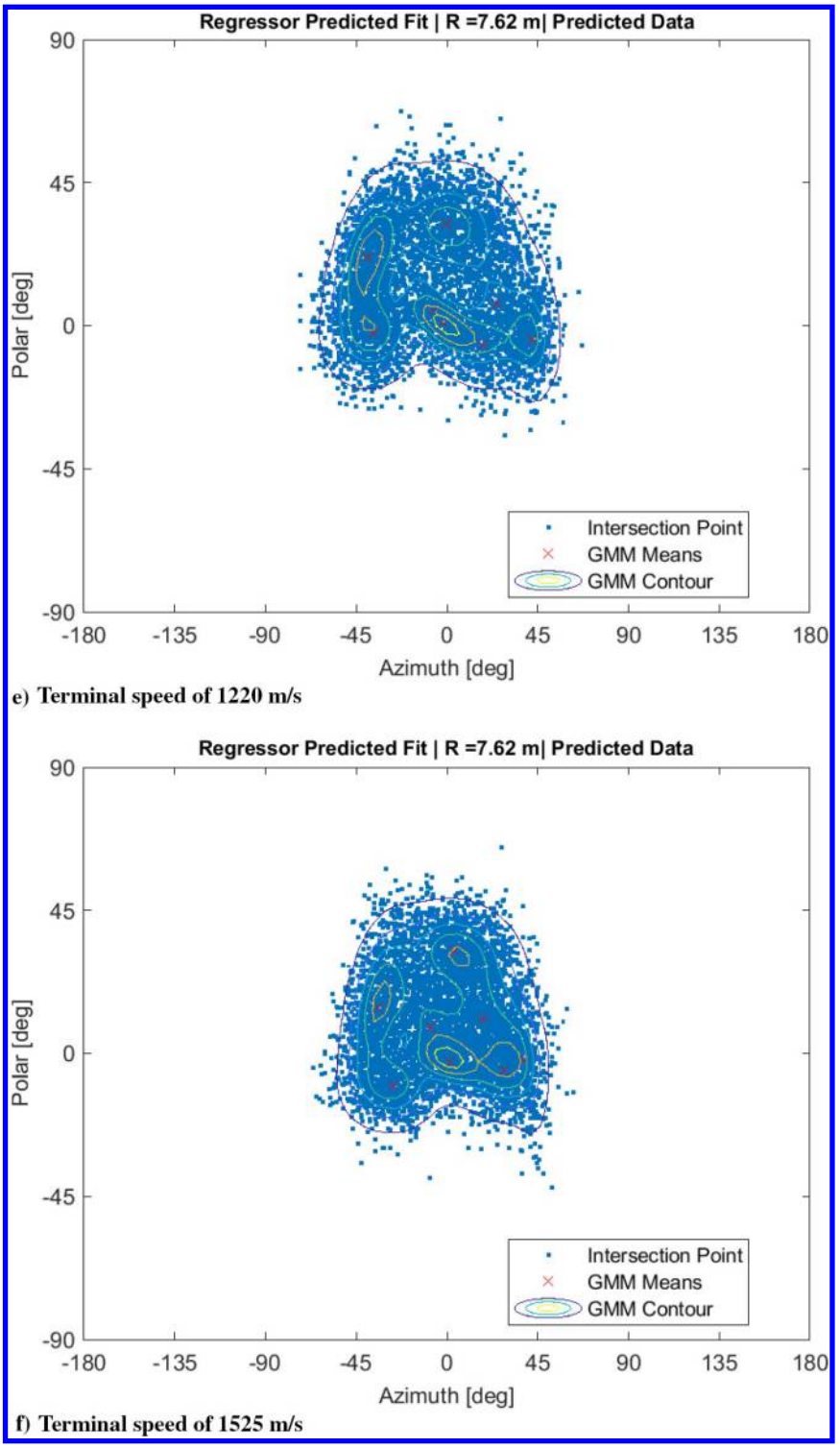

Fig. 15 (Continued).

predict the parameters for such a GMM. The model is merged with experimental data, and the performance of this assimilated model is evaluated via visual inspection of the features in experimental and simulation intersection distributions as well as the integration differences from a Monte Carlo simulation. It is shown that the assimilated model is able to learn the features of both the simulation and the experimental data. Although this ability to learn features of both datasets does not on its own validate the model, it does indicate that this new tool has utility in the warhead fragmentation realm. The model does in fact make predictions that make physical sense for dynamic detonations (i.e., distributions close in on the warhead direction of motion). It is expected that model predictions will improve as more experimental and simulation data are fed onto the training set. Although the ability to predict still lies within the confines of this initial rendition and scenario, it shows great promise as it can be expanded to predict more geometries such as mass and speed distributions.

Acknowledgments

This work was performed as a part of the Summer Faculty Fellowship Program for the U.S. Air Force Research Laboratory. The

investigations are supported through high-fidelity simulations by the U.S. Naval Air Warfare Center Weapons Division, the U.S. Air Force Office of Scientific Research (award number FA9550-20-1-0200), and the University of Florida through support for R. Bevilacqua during his research tenures at Eglin Air Force Base.

References

[1] Bogdanowicz, Z. R., Tolano, A., Patel, K., and Coleman, N. P., "Optimization of Weapon—Target Pairings Based on Kill Probabilities," *IEEE Transactions on Cybernetics*, Vol. 43, No. 6, 2013, pp. 1835–1844. https://doi.org/10.1109/TSMCB.2012.2231673

[2] Deep, K., and Pant, M., "Maximisation of Expected Target Damage Value," *Defence Science Journal*, Vol. 55, No. 2, 2005, pp. 133–139. https://doi.org/10.14429/dsj.55.1977

[3] Bogdanowicz, Z. R., and Patel, K., "Quick Collateral Damage Estimation Based on Weapons Assigned to Targets," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, Vol. 45, No. 5, 2015, pp. 762–769. https://doi.org/10.1109/TSMC.2014.2360823

[4] Humphrey, A., See, J. E., and Faulkner, D., "A Methodology to Assess Lethality and Collateral Damage for Nonfragmenting Precision-Guided Weapons," 2008.

[5] Gold, V. M., "Fragmentation Model for Large L/D (Length over Diameter) Explosive Fragmentation Warheads," *Defence Technology*, Vol. 13, No. 4, 2017, pp. 300–309. https://doi.org/10.1016/j.dt.2017.05.007

[6] Chu, E., Spencer, A., and He, J., "OpenCV and TYZX: Video Surveillance for Tracking," Sandia National Labs. Rept. SAND2008-5776, Albuquerque, NM, 2008, https://www.osti.gov/biblio/942060-opencv-tyzx-video-surveillance-tracking.

[7] Klare, M. T., "An 'Arms Race in Speed': Hypersonic Weapons and the Changing Calculus of Battle," *Arms Control Today*, Vol. 49, No. 5, 2019, pp. 6–13.

[8] Gold, V. M., Baker, E. L., and Pincay, J., "A Model for Fracture of Natural and Controlled Fragmentation Munitions," *SEM Annual Conference and Exposition on Experimental and Applied Mechanics 2007*, Vol. 1, Soc. for Experimental Mechanics, June 2007, pp. 1874–1879.

[9] Mott, N., "A Theory of the Fragmentation of Shells and Bombs," *Fragmentation of Ring and Shells*, Springer, New York, 2006, pp. 243–294. https://doi.org/10.1007/978-3-540-27145-1_11

[10] Gold, V. M., Baker, E. L., Poulos, W. J., and Fuchs, B. E., "PAFRAG Modeling of Explosive Fragmentation Munitions Performance," *International Journal of Impact Engineering*, Vol. 33, No. 1, 2006, pp. 294–304. https://doi.org/10.1016/j.ijimpeng.2006.09.032

[11] Gold, V. M., Baker, E. L., and Pincay, J. M., "Computer Simulated Fragmentation Arena Test for Assessing Lethality and Safety Separation Distances of Explosive Fragmentation Ammunitions," *WIT Transactions on Modelling and Simulation*, Vol. 45, May 2007, pp. 181–190. https://doi.org/10.2495/CBAL070171

[12] McGlaun, J. M., Thompson, S. L., and Elrick, M. G., "CTH: A Three-Dimensional Shock Wave Physics Code," *International Journal of Impact Engineering*, Vol. 10, No. 1, 1990, pp. 351–360. https://doi.org/10.1016/0734-743X(90)90071-3

[13] Krisko, P. H., "The New NASA Orbital Debris Engineering Model ORDEM 3.0," AIAA Paper 2014-4227, 2014. https://doi.org/10.2514/6.2014-4227

[14] Lemons, D. S., and Gythiel, A., "Paul Langevin's 1908 Paper 'On the Theory of Brownian Motion' ["Sur la Théorie du Mouvement Brownien," C. R. Acad. Sci. (Paris) 146, 530–533 (1908)]," *American Journal of Physics*, Vol. 65, No. 11, 1997, pp. 1079–1081. https://doi.org/10.1119/1.18725

[15] Baker, W. E., Dodge, F. T., and Westine, P. S., *Joint Munitions Effectiveness Manual*, U.S. Air Force, 1969.

[16] Angel, J., "Methodology for Dynamic Characterization of Fragmenting Warheads," Rept. ADA506417, 2009, p. 20, https://apps.dtic.mil/sti/citations/ADA506417.

[17] Halls, B. R., Quintana, E., Lebow, L., and Guildenbecher, D. R., "High-Speed Fragment Tracking with X-Ray Radiography," AIAA Paper 2019-0271, 2019. https://doi.org/10.2514/6.2019-0271

[18] Guildenbecher, D., Olles, J., Miller, T., Reu, P., Yeager, J., Bowden, P., and Schmalzer, A., "Characterization of Hypervelocity Fragments and Subsequent HE Initiation," Tech. Rept. SAND2018-7243C, 2019, https://www.osti.gov/biblio/1563131-characterization-hypervelocity-fragments-subsequent-he-initiation.

[19] Bishop, C. M., "Neural Networks," *Pattern Recognition and Machine Learning*, Springer, New York, 2006, pp. 225–290.

[20] Theodoridis, S., and Koutroumbas, K., *Pattern Recognition*, 4th ed., Academic Press, New York, 2008, 143–149.

[21] *MATLAB, Software Package, Ver. 9.5.0.944444 (R2018b)*, The Mathworks, Inc., Natick, MA, 2018.

[22] Anderson, T. W., and Darling, D. A., "Asymptotic Theory of Certain," *Annals of Mathematical Statistics*, Vol. 23, No. 2, 1952, pp. 193–212. https://doi.org/10.1214/aoms/1177729437

[23] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E., "Scikit-Learn: Machine Learning in Python," *Journal of Machine Learning Research*, Vol. 12, Nov. 2011, pp. 2825–2830.

[24] Gebru, I. D., Alameda-Pineda, X., Forbes, F., and Horaud, R., "EM Algorithms for Weighted-Data Clustering with Application to Audio-Visual Scene Analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 38, No. 12, 2016, pp. 2402–2415. https://doi.org/10.1109/TPAMI.2016.2522425

[25] Bishop, C., "Mixture Density Networks," Tech. Rept., Aston Univ., 1994, https://publications.aston.ac.uk/id/eprint/373/1/NCRG_94_004.pdf.

D. Livescu
Associate Editor