Student Works

8-2023

Predicting Dynamic Fragmentation Characteristics from High-Impact Energy Events Utilizing Terrestrial Static Arena Test Data and Machine Learning

Katharine Larsen

Riccardo Bevilacqua

Omkar S. Mulekar

Elisabetta L. Jerome

Thomas J. Hatch-Aguilar

Predicting Dynamic Fragmentation Characteristics from High-Impact Energy Events Utilizing Terrestrial Static Arena Test Data and Machine Learning

Katharine E Larsen[a], Riccardo Bevilacqua[a], Omkar S Mulekar[b], Elisabetta L Jerome[c], Thomas J Hatch-Aguilar[d]

[a]Embry-Riddle Aeronautical University, Daytona Beach, FL, 32114, USA
[b]University of Florida, Gainesville, FL, 32611, USA
[c]Air Force Test Center, Eglin Air Force Base, FL, 32542, USA
[d]Naval Air Warfare Center Weapons Division, China Lake, CA, 93555, USA

Abstract

To continue space operations with the increasing space debris, accurate characterization of fragment fly-out properties from hypervelocity impacts is essential. However, with limited realistic experimentation and the need for data, available static arena test data, collected utilizing a novel stereoscopic imaging technique, is the primary dataset for this paper. This research leverages machine learning methodologies to predict fragmentation characteristics using combined data from this imaging technique and simulations, produced considering dynamic impact conditions. Gaussian mixture models (GMMs), fit via expectation maximization (EM), are used to model fragment track intersections on a defined surface of intersection. After modeling the fragment distributions, k-nearest neighbor (K-NN) regressors are used to predict the desired characteristics. Using Monte Carlo simulations, the K-NN regression is shown to predict the distributions for both the total number of fragments intersecting a given surface, as well as the expected total fragment velocity and mass associated with that surface. This information can then be used to estimate the kinetic energy of the particle to classify the particle and avoid debris collisions.

Keywords: High-Impact Energy Events, Gaussian Mixture Models; K-Nearest Neighbors; Machine Learning; Prediction Methods; Regression

1. Introduction

Continual growth in space debris threatens the future of human interaction with space. As space operations and satellite activity grows, especially within the cislunar region, high-velocity debris collisions are more likely. Impact with already existing space debris has the potential to create more fragments and further pollute the space environment. Therefore, tracking and avoidance of current particles is imperative. While large debris pieces can currently be tracked, an understanding of fragment propagation characteristics following these collision events, including location, velocity, and mass, is increasingly important.

NASA's Standard Satellite Breakup Model (SSBM), or Standard Breakup Model (SBM), is one of the most widely used references to model the breakup of a satellite upon impact or explosion. Developed through theoretical and empirical methods, this SBM is based on both ground-based impact experiments and data from one on-orbit collision [1]. NASA's EVOLVE 4.0 implements this model and improved upon its previous iteration, which only considered fragments as spherical particles, underestimated the number of particles smaller than 10 cm, and often produced the same number of fragments for satellites of the same size [2]. However, EVOLVE is a one-dimensional, low earth orbit (LEO) only model. Following EVOLVE, NASA produced the LEO to GEO (geostationary orbit) Environmental Debris model (LEGEND) to model past and future breakup events in three-dimensions and within a greater boundary region [1,3,4]. These models need consistent update to improve the model and include new data, collected from ground-based experimentation in addition to changes in the space environment.

The United States military has worked with NASA, and other partners, to continually update the SBM. One well known project is DebriSat, used to update the model with more current satellite qualities such as structure, material, and technology [5,6]. DebriSat was a ground-based experiment used to model catastrophic breakup of payloads in the LEO environment and included the detection, extraction, and characterization of fragments.

Following a series of documents recognizing space debris as a problem, the Department of Defense (DoD) has also developed their own Orbital Debris Program, beginning in the late 1980's [7]. Since then, the DoD has conducted numerous terrestrial tests using weapons analysis methods to represent orbital debris modeling. One method for data collection included arena testing of a warhead. Then, using velocity screens, collection bundles, time-of-arrival pins, x-rays, high-speed cameras, and pressure and strain gauges, the fragmentation was characterized by recording fragment material, size, velocity, and spatial distribution [7]. The research presented in this paper uses a similar approach, using data from static arena test detonations with more modern collection methods, such as stereoscopic imaging, in addition to the standard approach.

Through Wright-Patterson Air Force Base, the US military has also performed research to assess and prevent the damage of aluminum aircraft structures when impacted by steel fragments [8]. Using imaging techniques such as high-speed cameras and lasers from the Air Force Institute of Technology, this project looked more at the vulnerability of the structure to impact than focusing on fragmentation itself.

Another previous debris model, Fragmentation Algorithms for Strategic and Theater Targets (FASTT) produces estimates for fragment characteristic distributions of different aerospace structures, in addition to ground-based collisions [9]. Most of the distributions developed are for collision distributions and velocity distributions resulting from an explosive event is listed as a future goal for this algorithm.

There have also been methods proposed to model debris interaction with satellites using structural analysis. For example, a Collision Simulator Tool (CST) was developed to model objects using a mesh structure or a net of macroscopic elements, before simulating the model, and predicts events in relation to the catastrophic threshold [10]. The term catastrophic refers to the complete fragmentation of both impacting objects. A collision becomes catastrophic when the kinetic energy of the smaller object, or projectile, divided by the target's mass is greater than 40 J/g [2]. While small fragments may not cause catastrophic damage, they do have the potential to damage necessary systems of a space vehicle. Another recent study looks to machine learning techniques to classify fragments. They propose a neural network augmented density-based spatial clustering of applications with noise (DBSCAN) to investigate debris of unknown origin and assign them debris families, where each debris family is assigned to a specific breakup event [11].

This presented paper, like the previously stated classification method, utilizes machine learning methods to predict characteristics of fragments produced from high-velocity impact events or explosions, either intentional or non-intentional. This research uses terrestrial data collected using a stereoscopic imaging method described in the following subsection. In lieu of space debris data with as much accuracy for individual fragment tracks, and with these recent improvements in terrestrial static arena testing, this research uses static arena test data to validate the proposed method. Not only can this method be used to predict large fragments, but also small, usually undetectable, fragments, to determine encounters and prevent future collisions.

1.1. Terrestrial Fragmentation Background

The mid-1940's brought one of the first and most prominent fragmentation theories – Mott's formula. Mott presented a formula designed to predict the distribution of metal shell casing fragments from an explosive event, highly dependent on the structure and material of a metal casing [12]. While this formula can represent a large distribution of the casing fragmentation, Mott made data distribution assumptions, giving it only a generic representation of the center of spatial distribution, where most of the fragments lie, often excluding smaller fragments and those existing outside the center of the distribution [13]. Additionally, the formula must be adjusted for different pre-detonation properties (e.g., material), affecting the accuracy. The machine learning technique proposed in this paper aims to remove these distribution assumptions using gaussian mixture models (GMMs) and create a more generic procedure that can be readily used for various pre-detonation conditions, including terminal velocity and orientation.

Following Mott's formula came many other fragmentation methods. One paper, published in 2009, presents a comparison of seven different theoretical mass distribution models: the Mott, the generalized Mott, the Grady, the generalized Grady, the log-normal, the Weibull, and the Held [14]. Comparing all methods to experimental data sets, the investigators found the generalized Grady distribution fit best to their experimental cases. These methods, like Mott's formula, are two-dimensional and may depend on the impact scenario and material for accuracy.

Around the same time as Mott's formula, Gurney produced a formula to estimate the initial velocities of the fragments produced by an explosion. Again, this equation is highly dependent on the casing and explosive characteristics, such as type and quantity. Like Mott, Gurney also made assumptions when producing this formula. One major assumption of Gurney's equation is that the total kinetic energy per unit mass of explosive is independent of the fragment characteristics such as size [15]. Therefore, each fragment produces the same initial velocity for all fragments. Additionally, Gurney's equation is one-dimensional and requires on-axis initiation for cylindrical explosives.

Using distribution theories, described above, and finite element analysis (FEA), modeling and simulations have since been developed to predict fragment characteristics and decrease the financial and labor costs of experimental testing. One method uses a high-rate finite difference computer program, known as CALE, to predict numerical models of fragment spray, including mass and velocity distributions [16.17].

Following this work came another fragmentation computer code, known as Picatinny Arsenal FRAGmentation (PAFRAG). This method is primarily based on Mott's theory, but the choice of fragmentation theory can be altered [18,19]. Using CALE and PAFRAG, Picatinny Arsenal integrates analytical and experimental techniques by combining high-strain/high-strain-rate computer modeling with semi-empirical modeling and experimentation to estimate lethality and a safe separation distance [20,21].

Similar fragmentation work has begun using Artificial Neural Networks (ANNs) to predict rock wall fragmentation [22]. One study compares rock fragmentation prediction methods using linear and non-linear regression to ANN, noting that predictions were more accurate using ANN [23]. This can be attributed to the rigidity of linear and non-linear regression methods, often eliminating some data with multiple inputs and outputs. For the same reason, regression methods other than linear and non-linear regression are chosen for the proposed research here.

In recent years, stereoscopic imaging has become an addition to the classic static arena test. High-speed stereoscopic imaging utilizes still images from multiple different camera angles to provide two-dimensional images with depth, essentially giving the image three-dimensional qualities [24]. Eglin Air Force Base, as well as other testing bases, now uses a stereoscopic vision system, known as the Optical Warhead Lethality Sensor Suite (OWLSS), which measures position vector tracking for individual fragments [26]. From the position vector information, velocity vectors can be calculated.

This paper expands and improves initial ideas proposed in 2021 [27]: predicting fragmentation characteristics using GMMs to generate training data from both static arena tests and simulations, and random forest regression (RFR) to make fragment count predictions from the generated training data. Unlike this previous study, this paper improves upon the fragment spatial distribution predictions performed previously and lays the groundwork for fragment velocity and mass predictions. Using static arena test data from the U.S. Naval Air Warfare Center's Weapons Division (NAWCWD) at China Lake and simulation data developed using the initial files of the static arena tests and provided by NAWCWD fragment flyout research software, this research builds on the techniques proposed using multivariate GMMs, expectation maximization (EM) to provide a weighted values to velocity and mass GMMs, and k-nearest neighbors (K-NN) regression to predict the spatial, velocity, and mass distributions of high-velocity fragments. Accordingly, this research aims to develop a model to predict dynamic fragmentation fly-out behavior from high-velocity impact events when only static experimental data is available. In the future, this model will be used to obtain new relationships connecting the system's characteristics before detonation to post-detonation characteristics, in terms of fragment spatio-temporal distribution, and energy.

Before presenting the process used to predict the fragment characteristics, Section II presents the research objectives. Section III presents a description of the data available for this research. Section IV reviews materials and machine learning techniques utilized. The methodology is proposed in Section V and the results, validation, and evaluation of results are in Section VI. Finally, Section VII consists of conclusions and suggestions for future work.

2. Research objectives

Given impact conditions, i.e., the attitude and velocity at point of impact, and a chosen radius of intersection, this research will provide more accurate fragmentation predictions than previous methods. The overall goal is to create a transfer function to predict in-flight fragmentation behavior, specifically fragment distribution, velocity, and fragment mass, over time using machine learning techniques, static arena test data, and corresponding high-fidelity numerical simulations from NAWCWD. With this information, one can estimate the kinetic energy of debris particles. Fig. 1 displays a diagram of the inputs and outputs of the proposed machine learning model.

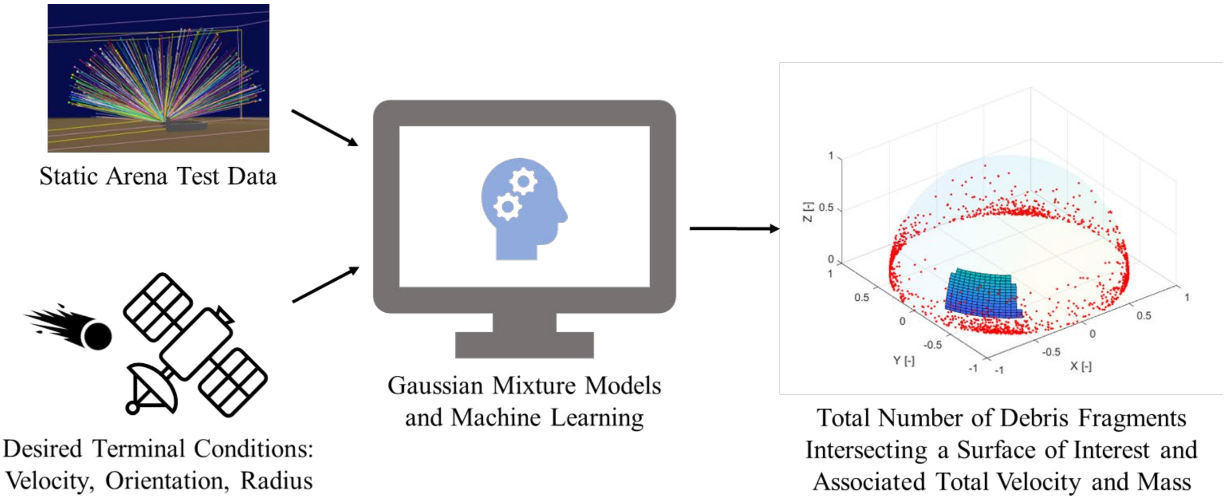

Static Arena Test Data
Desired Terminal Conditions:
Velocity, Orientation, Radius
Gaussian Mixture Models
and Machine Learning
Z [-]
1
0.5
0
1
0.5
0
-0.5
-1
Y [-]
-1
-0.5
0
0.5
1
X [-]
Total Number of Debris Fragments
Intersecting a Surface of Interest and
Associated Total Velocity and Mass

Fig. 1. The inputs and outputs of the desired model.

3. Description of available data

Most data available for this research consists of experimental static arena test data and simulation data developed using experimental data as initial conditions [27]. The experimental data comes from static pipe bomb detonations performed by the NAWCWD at China Lake. Using stereoscopic tracking systems, the pipe bomb detonations were performed for 3 different cases, 2 mm, 5 mm, and 7 mm ball bearings. An example of the experimental setup for the 5 mm static detonation is shown in Fig. 2, which also displays the radially symmetric placement of the ball bearings. Therefore, this research assumes there is a symmetric pair across the y-axis, making a duplicated position at $[x, -y, z]$ for every fragment at $[x, y, z]$. The fragment tracks for the 5 mm ball bearing case can be referenced in Fig. 3 and Fig. 4.

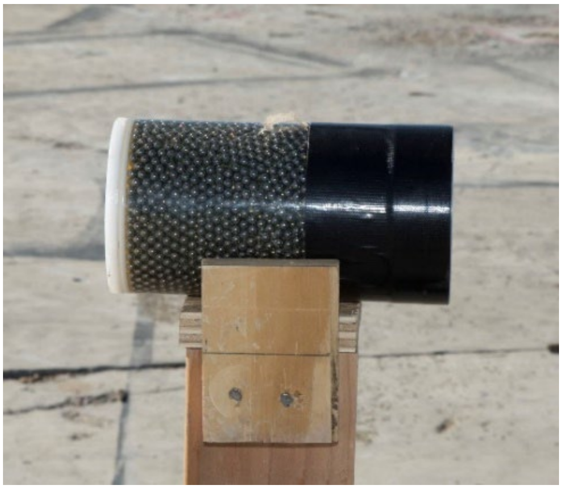


Fig. 2. The experimental set-up for the 5 mm ball bearing static arena test.

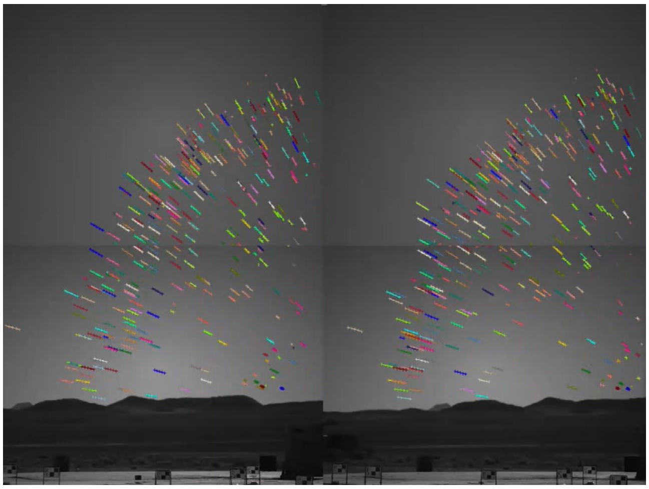


Fig. 3. Fragment tracks from the stereoscopic imaging.

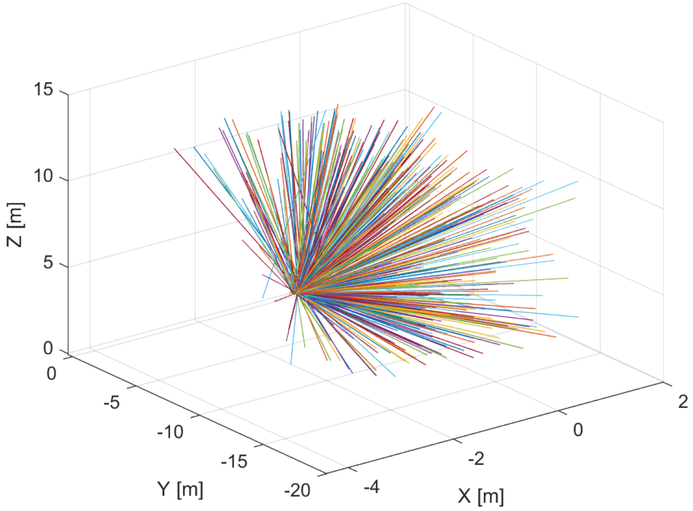


Fig. 4. Tracks collected from the 5 mm ball bearing static arena test.

Using the 5 mm ball bearing static experimental case, NAWCWD provided dynamic simulation data by iteratively solving Langevin's equations, stochastic differential equations to describe the Brownian particle motion over time [28,29]. NAWCWD simulated time history tracks for all 3861 ball bearing fragments for 1100 different cases at various terminal speeds and detonation orientations, displayed in Table 1, making up approximately 1.3 terabytes of data. It should be noted that these simulations did not consider various types of targets, but rather impact with a flat surface. In this case, it was considered to be a terrestrial impact colliding with the ground. An example of the fragment tracks for the simulation case with 0° roll, pitch, and yaw and a terminal velocity of $0\ \mathrm{m/s}$ are displayed in Fig. 5.

Table 1. 5 mm ball bearing NAWCWD simulation cases.

Variables		Range	Number of Cases	Units
Terminal Speed	Magnitude	0, 152, 304, 457, 609, 762, 914, 1066, 1219, 1371, 1524	11	m/s
Orientation of at Burst Point	Pitch	-90, -60, -30, 0	4	degrees
	Yaw	-60, -30, 0, 30, 60	5	
	Roll	0, 45, 90, 135, 180	5	
Total Number of Simulations			1100 (1.3 TB)	

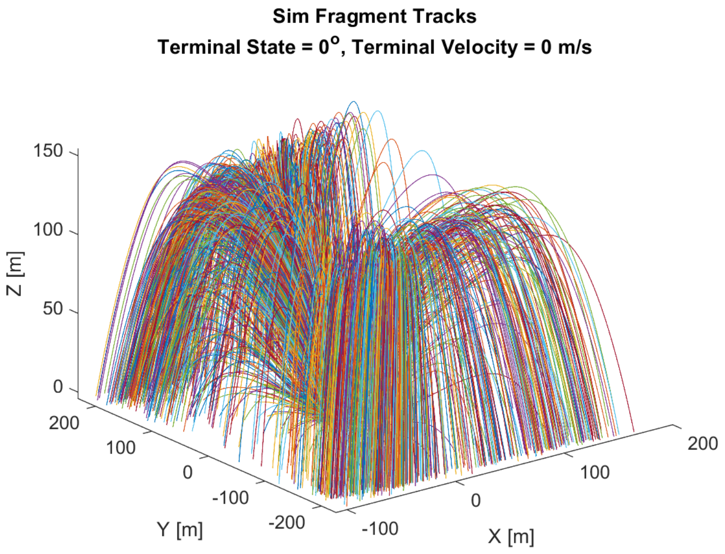


Fig. 5. Fragment tracks from one of the NAWCWD simulations.

By combining the limited experimental data with the larger set of simulation data, static and dynamic characteristics can be included in the machine learning model. However, as discussed previously, there are limitations with current technology [27]. Over time, future data collected under improved systems will enhance the modeling process, by improving the quality, as well as the amount, of the data.

In addition to the ball bearing data, the NAWCWD also provided experimental and simulation data for a naturally fragmenting system, collected under the same procedure. The simulation data is displayed in Table 2, making up 1000 simulation cases. The fragment tracks for a simulation case with 0° roll, pitch, and yaw and a terminal velocity of $0\ \mathrm{m/s}$ are displayed in Fig. 6.

Table 2. Naturally fragmenting simulation cases.

Variables		Range	Number of Cases	Units
Terminal Speed	Magnitude	0, 152, 305, 457, 610, 914, 1219, 1524	8	m/s
Orientation of at Burst Point	Pitch	0, 22.5, 45, 67.5, 90	5	degrees
	Yaw	-45, -22.5, 0, 22.5, 45	5	
	Roll	0, 22.5, 45, 67.5, 90	5	
Total Number of Simulations			1000	

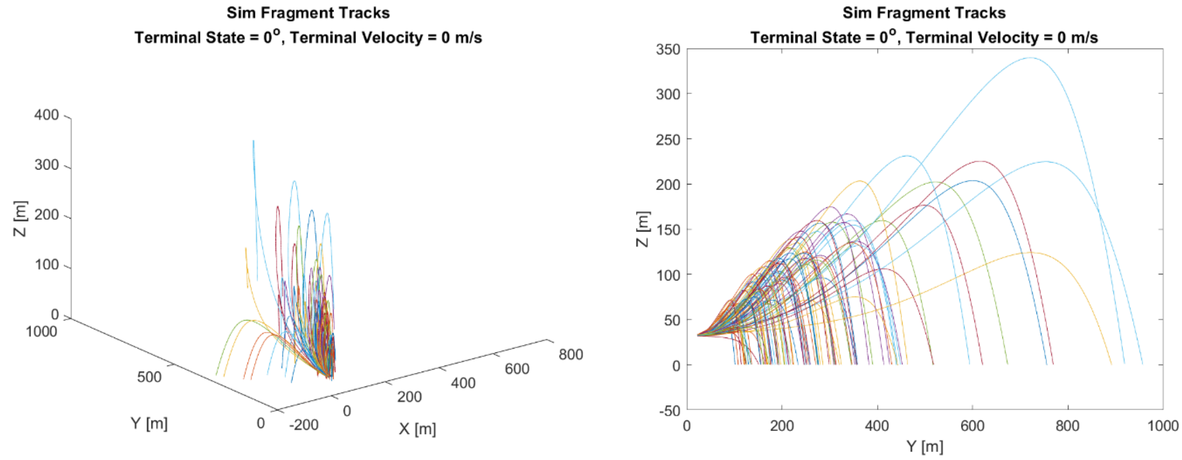


Fig. 6. Fragment tracks from one of the realistic NAWCWD simulations.

4. Overview of utilized techniques

The following section outlines materials and machine learning methods that this research utilizes.

4.1. Software

The primary software in this research includes MATLAB and python. The data extraction and training data preparation are performed in MATLAB and most of the regression learning is performed in python, where Scikit-Learn and other machine learning libraries can be utilized [30].

4.2. Gaussian mixture models

Training sets can be generated by representing the available data using probability distributions. In the case of fragmentation, these distributions can be used to describe fragment track intersections on surfaces. This surface can be created by finding the points of intersection along a sphere with a chosen radius of intersection and converting from cartesian coordinates to a polar-azimuth plane. An example is displayed in Fig. 7 for a simulation with an intersection radius of 75 m.

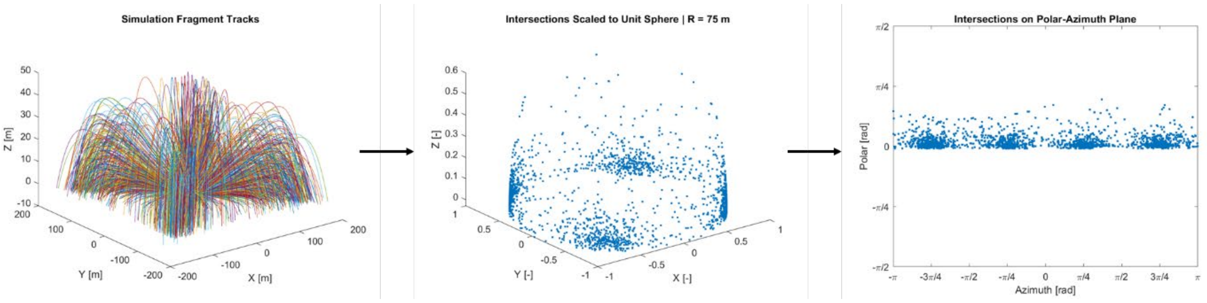


Fig. 7. Converting from cartesian coordinates to a polar-azimuth plane for a simulation with R = 75 m.

To convert the cartesian coordinate system of $[x, y, z]$ to polar and azimuth angles, ϕ and θ respectively, the following equations are used

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

$$\theta = tan^{-1}\left(\frac{y}{x}\right) \tag{2}$$

The multivariate gaussian distribution is defined as

$$\mathcal{N}(\vec{X}|\vec{\mu},\Sigma) = \frac{1}{\sqrt{(2\pi)^{dim}|\Sigma|}} exp\left(-\frac{1}{2}(\vec{X}-\vec{\mu})^T \Sigma^{-1}(\vec{X}-\vec{\mu})\right) \tag{3}$$

where $\vec{X}$ is the random variable vector, Σ is a symmetric covariance matrix, μ is a vector containing the means, and dim is the dimension of the dataset [31,32]. However, these distributions are unimodal and do not capture the full shape of the intersection distribution, as they only represent one subpopulation of the data.

Instead, gaussian mixture models (GMMs) are a weighted linear combination of multivariate gaussian distributions, ultimately combining the subpopulations to represent the entire population. They have three defining parameters: the mean, μ, represents the location of each mode of the distribution, or subpopulation center, the covariance, Σ, represents the spread of the data distribution, and mixing coefficients, π_i. In the multivariate case the GMM is the probability distribution defined as

$$p(\vec{X}) = \sum_{i=1}^{N} \pi_i \mathcal{N}(\vec{X}|\vec{\mu}_i, \Sigma_i) \tag{4}$$

where N is the number of components, $\sum_{i=1}^{K} \pi_i = 1$ where π_i are the mixing coefficients, and $\mathcal{N}(\vec{x}|\vec{\mu}_i, \Sigma_i)$ is the multivariate distribution described in Equation 3 [31,32].

Using GMM parameters, the model can be tuned further to optimize the fit of the data and prevent overfitting or underfitting. These parameters include the number of components and the covariance type, whether it be full, tied, diagonal, spherical, etc. Previously, 4 different cases were tested: full covariance with 8 components, full covariance with 4 components, diagonal covariance with 8 components, and diagonal covariance with 4 components [27]. It was concluded that GMMs with 8 components and a full covariance perform better than the others. Therefore, this paper focuses on the full covariance type with 8 components.

4.3. Expectation maximization for weighted data

EM is an iterative method used to calculate the maximum likelihood estimates (MLE) of GMM parameter estimates, essentially assigning weights, or cluster numbers, to the points of the dataset. This value can then be used to represent the velocity of each point. Then, each point can be treated different than the next and both the position and velocity can be taken into consideration for the velocity distribution predictions. This method can also be used to consider the mass of a particle as a cluster number.

EM consists of two main steps: the E-Step and the M-Step. In the E-Step, or expectation, the posteriors are calculated as

$$\eta_{ik}^{(r+1)} = \frac{\pi_k^{(r)} \mathcal{N}\left(X;\mu,\frac{1}{w}\Sigma\right)}{\sum_{k=1}^{K} \pi_k \mathcal{N}\left(X;\mu_k,\frac{1}{w}\Sigma_k\right)} \tag{5}$$

where K is the number of mixture components and w is the weight defined by

$$w_i = \sum_{j \in S_i^q} exp\left(-\frac{d^2(x_i,x_j)}{\sigma}\right) \tag{6}$$

where $d(x_i, x_j)$ is the Euclidean distance, S_i^q is the set containing q nearest neighbors of the query point, x_i, and σ is a positive scalar [33].

In the M-Step, or maximization, the mixing coefficients, means, and covariances are calculated as

$$\pi_k^{(r+1)} = \frac{1}{n}\sum_{i=1}^{n} \eta_{ik}^{(r+1)} \tag{7}$$

$$\mu_k^{(r+1)} = \frac{\sum_{i=1}^{n} w_i \eta_{ik}^{(r+1)} x_i}{\sum_{i=1}^{n} w_i \eta_{ik}^{(r+1)}} \tag{8}$$

$$\Sigma_k^{(r+1)} = \frac{\sum_{i=1}^{n} w_i \eta_{ik}^{(r+1)} (x_i - \mu_k^{(r+1)}) \left(x_i - \mu_k^{(r+1)}\right)^{\mathsf{T}}}{\sum_{i=1}^{n} \eta_{ik}^{(r+1)}} \tag{9}$$

The E-Step and the M-Step are repeated until convergence of the GMM parameters or when a maximum number of iterations is reached. Convergence occurs when the difference between the log likelihood of two consecutive iterations is within a given tolerance, or when the maximum number of iterations has been reached [33].

4.4. Random forest regression

Following the development of training data, regression tools can be used to learn a relationship between input data (independent variables) and output data (dependent variables). In previous methods, random forest regression (RFR) was a suitable machine learning tool, noting multiple decision trees as a method to avoid overfitting data [27]. However, RFR does not produce good velocity predictions. Additionally, while it was suitable for the fragment count predictions, predictions were improved when using k-nearest neighbors (K-NN) regression learning.

When using multiple trees, RFR can be described as the mean of all the individual trees

$$g(u) = \frac{1}{M}\sum_{i=1}^{M} f_i(u) \tag{10}$$

where f_i represents each individual tree, M is the number of trees in the random forest, and u is a regressor input vector [27,34].

4.5. K-nearest neighbors regression

However, while RFR was suitable for the fragment count predictions, other regression methods were considered. Predictions were improved for both the count predictions and the velocity predictions when using k-nearest neighbors (K-NN) regression learning. The goal of the K-NN algorithm, as well as other regression methods, is to create a function to describe an input-output relationship. K-NN, like RFR, uses an average of observations to produce a model. However, unlike RFR, K-NN takes an average of the closest points based on a local neighborhood. K-NN computes the mean as

$$f_{knn}(x_q) = \begin{cases} \frac{\sum_{i=1}^{k} w_i f(x_i))}{\sum_{i=1}^{k} w_i}, & D(x_q, x_i) \neq 0 \\ f(x_1), & D(x_q, x_i) = 0 \end{cases} \tag{11}$$

where x_q is the query point, x_i is a close point, or a neighbor, w is the weight defined as

$$w_i = \frac{1}{D(x_q, x_i)} \tag{12}$$

and k is the size of the neighborhood [35,36]. In the case that the minimum distance between the pair of objects is zero, $D(x_q, x_i) = 0$, the weight is undefined. Instead of a using the weighted average, the corresponding output training data of the single closest point is used.

The size of the neighborhood, k, is a chosen integer, and must be adjusted to avoid overfitting and underfitting. For example, if one were to pick $k = 1$, the output would result in the parameters of the most similar point in the training set. In a binary case, where there are only two options for the output variable, this would be appropriate. Otherwise, there is a risk of underpredicting or overpredicting.

Euclidean distance is a common method for calculating the distance between points in a dataset. Rather than calculating a Euclidean distance, Mahalanobis distance works well for correlated multivariate data and solves for the difference between multiple variables or dimensions and can be calculated as such

$$D(x_q, x_i) = \sqrt{(x_q - x_i)^T C^{-1}(x_q - x_i)} \tag{13}$$

where x_q and x_i represent a pair of objects and C is the sample covariance [37].

4.6. Boundaries of interest

After predicting the total fragment counts, fragment velocities, and their respective distributions, fragment characteristics within a given boundary of interest can be calculated using double integrals defined as

$$Fragment\ Count = N_{total} \iint_S \ p_N(x)\, dS \tag{14}$$

$$Fragment\ Velocity = V_{total} \iint_S \ p_V(x)\, dS \tag{15}$$

where N_{total} is the predicted total number of fragments passing through the entire sphere of intersection, V_{total} is the predicted total velocity, or the sum of all fragment velocities intersecting the entire sphere of intersection at the chosen radius, and dS is described as

$$dS = d\phi d\theta \tag{16}$$

where ϕ is the polar angle and θ is the azimuth angle. An example of this boundary can be seen in Fig. 8, below.

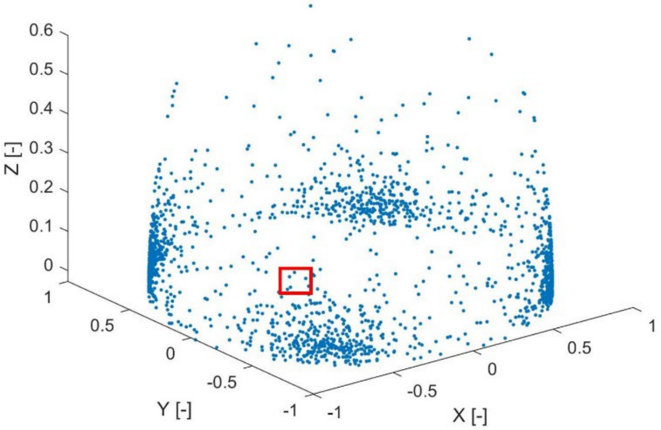


Fig. 8. An example of a chosen region of interest on an intersection radius scaled to a unit sphere.

5. Methodology

Using the provided static arena test data and corresponding simulation data, a training pool is generated as probability distributions for fragment track intersections at various intersection radii mapped on a polar-azimuth plane. K-NN regression uses an average of the nearest neighbors to predict the GMM parameters of a point of interest. Given an input as a vector, including the terminal state (impact velocity and orientation as roll, pitch, and yaw) and an intersection radius, the GMM parameters of a probability distribution can be predicted, using two separate K-NN regressors. One regressor predicts the fragment count distribution shown in Fig. 9 and the other predicts the fragment velocity distribution shown in Fig. 10. In addition to the GMM parameters, the total fragment counts and total fragment velocities are also output for the entire sphere of intersection, which can then be fed into the respective double integrals to find totals within a chosen area of interest. Using these values, the average velocity can be calculated using the process displayed in Fig. 11.

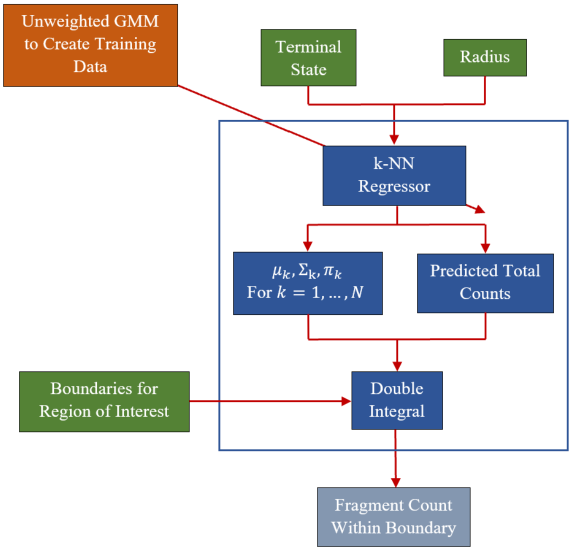

Unweighted GMM to Create Training Data
Terminal State
Radius
k-NN Regressor
μ_k, Σ_k, π_k For $k = 1, \ldots, N$
Predicted Total Counts
Boundaries for Region of Interest
Double Integral
Fragment Count Within Boundary

Fig. 9. Diagram of the fragment count predictor.

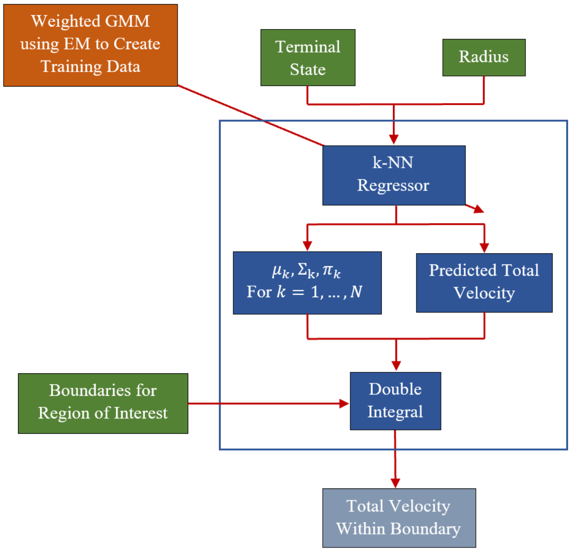


Fig. 10. Diagram of the fragment velocity predictor.

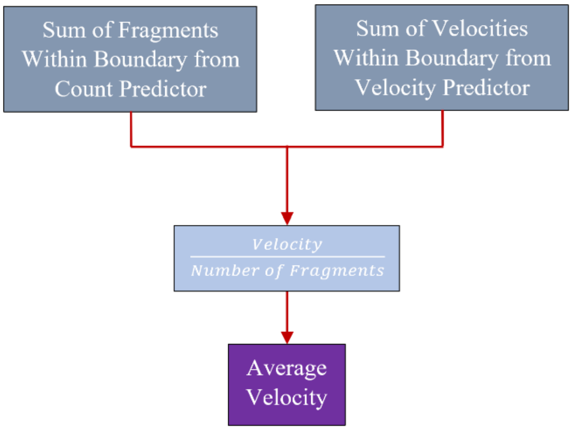

Sum of Fragments Within Boundary from Count Predictor
Sum of Velocities Within Boundary from Velocity Predictor
$\frac{Velocity}{Number\ of\ Fragments}$
Average Velocity

Fig. 11. Diagram of the average velocity within a given boundary of interest.

5.1. Generating training data

For a range of intersection radii, in this paper 65 radii ranging from 7.62 m to 182.88 m, polar-azimuth maps are generated. An example of this process is shown in Fig. 12, Fig. 13, and Fig. 14 for a NAWCWD simulation case with an impact orientation of 135° roll, 60° pitch, and −60° yaw and an impact velocity of 0 m/s. For each map, training data is generated as probability distributions using gaussian mixture models. Training data sets for fragment counts and fragment velocities are generated for both the experimental static arena test data and the corresponding 1100 NAWCWD simulations.

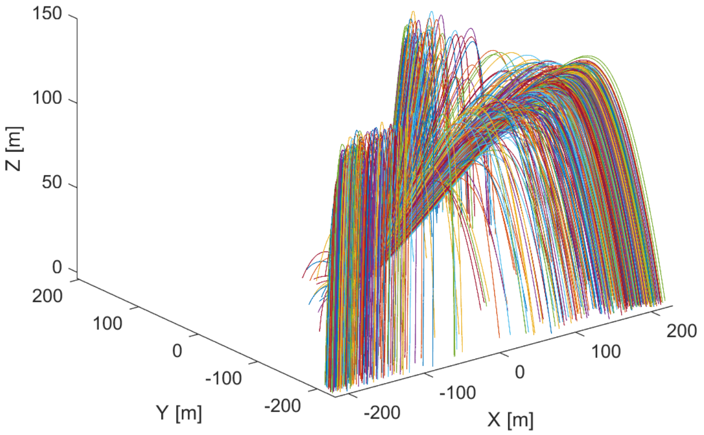


Fig. 12. Fragment tracks from one of the NAWCWD simulations.

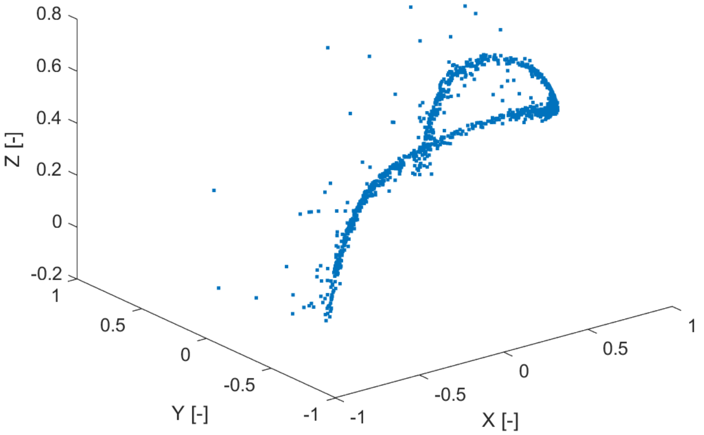


Fig. 13. Fragments intersecting a radius of ~71.65 m on a unit sphere from one of the NAWCWD simulations.

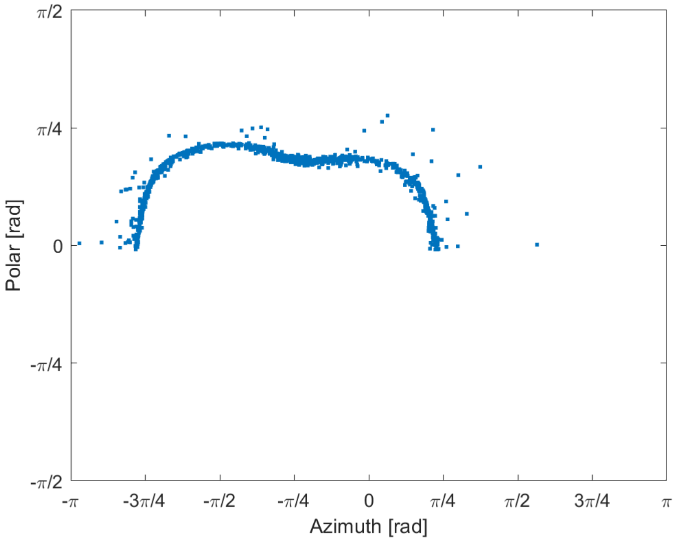


Fig. 14. Fragments intersecting a radius of ~71.65 m mapped onto a polar-azimuth coordinate system.

As mentioned previously, a paper compared four cases of GMMs finding that GMMs with 8 components and full covariance are best suited for this data [27]. They also note the importance of including both the experimental static arena test data and the simulation data in the training data. Although the simulation data makes up a large percentage of the training data population, the experimental data should be included to incorporate realistic fragment dynamics, not experienced by the simulations. Additionally, by including both cases, some regions excluded by the experimental case are included by the simulation cases and vice versa.

The fragment count training data is generated using the described GMMs, but the fragment velocity training data is generated using GMMs fit via expectation maximization. During this process one should also note the importance of proper data formatting. Using random initialization, the means, or centroids, of the dataset distributions may not converge, as EM is highly dependent on its initialization point. To preserve the component ordering for similar distributions, the EM algorithm should be strictly initialized using the GMM without EM.

5.2. Training the model

Once training data is produced, the model can be trained. Using K-NN regression, count predictions are generated using the training data from GMMs without EM and fragment velocity predictions are generated using the training data from GMMs with EM. After predicting the total fragment counts and fragment velocities and their respective distributions, fragment characteristics within a given boundary of interest can be calculated using the double integrals described previously, in Equation 14 and Equation 15.

5.3. Validation Methods

After training the model, validation methods can be used to verify and improve the process. This research uses Monte Carlo simulation as the primary validation method. Monte Carlo simulations artificially model data using random inputs [38]. This random generation enables fast predictions to evaluate the full model. In this research the randomly generated parameters consist of the simulation number, with an associated terminal attitude and velocity, and radius of intersection. Using these parameters and a randomly generated polar-azimuth region of interest, the total fragments within the given boundary from both the direct fragment counts and predicted fragment counts are found and compared using the method proposed in Fig. 15. This same method can be used for the velocity comparison. A diagram of this proposed model is displayed in Fig. 16, with the only difference from the proposed model in Fig. 15 being the addition of the fragment counts from each case to observe the differences average velocity as well as the total velocity within the boundary of interest. These differences are produced for multiple runs at various randomized conditions to evaluate the model.

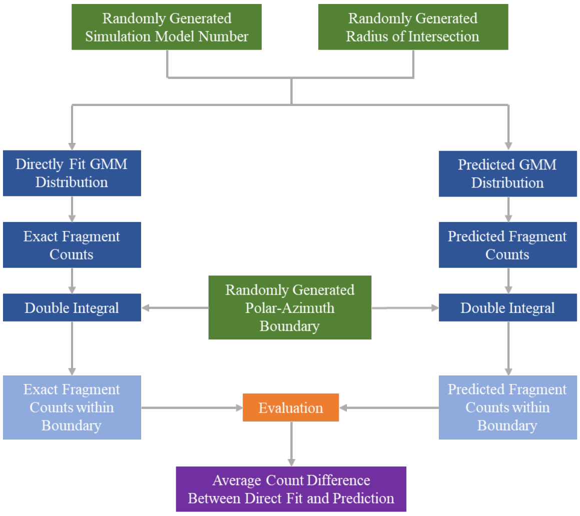

Randomly Generated Simulation Model Number
Randomly Generated Radius of Intersection
Directly Fit GMM Distribution
Predicted GMM Distribution
Exact Fragment Counts
Predicted Fragment Counts
Randomly Generated Polar-Azimuth Boundary
Double Integral
Double Integral
Exact Fragment Counts within Boundary
Evaluation
Predicted Fragment Counts within Boundary
Average Count Difference Between Direct Fit and Prediction

Fig. 15. Diagram of the Monte Carlo validation method for count comparison.

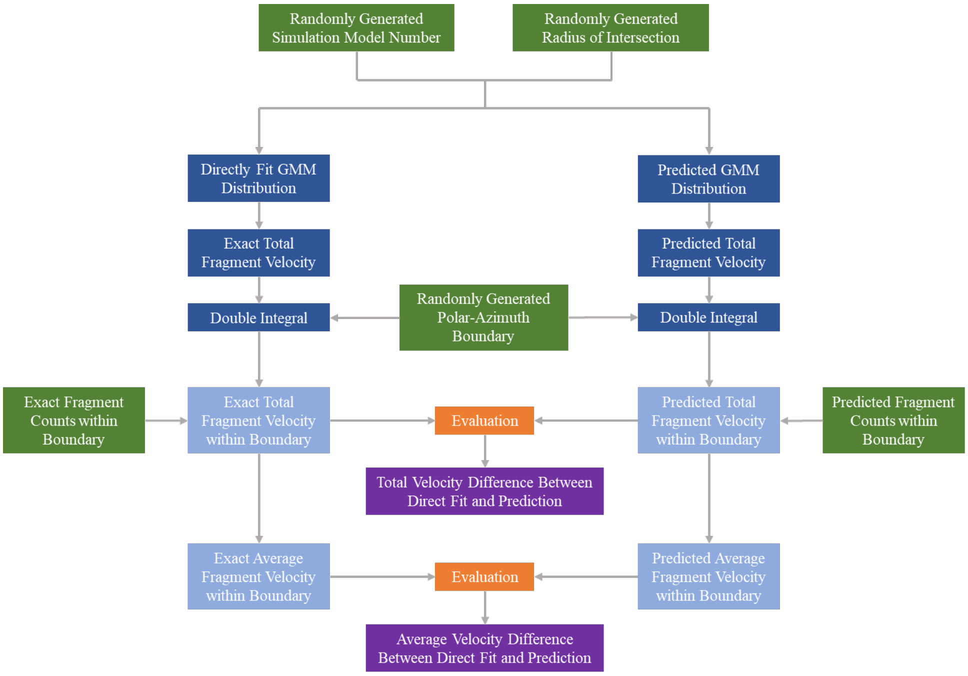

Randomly Generated Simulation Model Number
Randomly Generated Radius of Intersection
Directly Fit GMM Distribution
Predicted GMM Distribution
Exact Total Fragment Velocity
Predicted Total Fragment Velocity
Randomly Generated Polar-Azimuth Boundary
Double Integral
Double Integral
Exact Fragment Counts within Boundary
Exact Total Fragment Velocity within Boundary
Evaluation
Predicted Total Fragment Velocity within Boundary
Predicted Fragment Counts within Boundary
Total Velocity Difference Between Direct Fit and Prediction
Exact Average Fragment Velocity within Boundary
Evaluation
Predicted Average Fragment Velocity within Boundary
Average Velocity Difference Between Direct Fit and Prediction

Fig. 16. Diagram of the Monte Carlo validation method for velocity comparison.

6. Results and analysis

Previous work suggested RFR as a suitable method to predict the number of fragments but did not predict the fragment velocities or fragment masses [27]. The research presented here includes random forest regression as a mean of comparison, but considered other regression techniques, finding k-nearest neighbors regression with $k = 2$ to yield the best results for both the fragment velocity and the fragment count predictions. Using the Monte Carlo validation, discussed in the previous section, both techniques were analyzed over 1000 simulations, each with 20 different randomized polar-azimuth regions, creating 20,000 randomized runs. The mean and standard deviation of the fragment count differences for both regression methods are provided in Table 3. Table 4 displays results from the total and average fragment velocity predictions. In addition to comparing the two different regression techniques for the velocity prediction, Table 4 also compares velocity predictions from weighted velocity training data and unweighted training data. The weighted training data is generated using EM, while the unweighted training data has the same GMM parameters as the count data. By making this comparison, one can see how using EM affects the predictions.

Table 3. Mean and standard deviation of fragment count differences.

Regression Technique	Mean Count Difference, Fragments	Standard Deviation, Fragments
Random Forest	2.7	37.5
K-NN	1.8	28.0

Table 4. Mean of fragment velocity differences.

Regression Technique	Mean Total Velocity Difference, m/s	Mean Average Velocity Difference, m/s
RFR	2774.0	-66.7
RFR with Count Data	3212	-75.5
K-NN	904.3	31.9
K-NN with Count Data	2087.6	-4.58

In both the count and velocity predictions, K-NN proves to be a more accurate regression strategy. However, as different datasets are used in the future, other regression methods should still be under consideration, as the methodology selection is reliant on the dataset and its characteristics. In this case, K-NN suited the data's structure well and the metric, Mahalanobis distance, chosen to determine distance between a query point and its neighbors was appropriately selected. Additionally, it was a faster tool than RFR.

Using the K-NN regression technique, and the weighted velocity dataset, the average kinetic energy was also calculated using the results from the Monte Carlo simulations, as the fragment mass is not uniformly distributed. The mean difference between the predicted average kinetic energy of each fragment is 7.281 J.

The estimated fragment count and velocity distributions for a case generated by the Monte Carlo simulations, with 0° roll, 0° pitch, 30° yaw, and 914.4 m/s terminal conditions, is displayed in Fig. 17 and Fig. 18. These figures show the GMM distributions directly fit to the data compared to the predicted GMM distributions produced by the machine learning model. The counted fragments across the entire surface of intersection for this case is 1,943 fragments and the model predicted a total of 1,942 fragments, yielding only an absolute difference of 1 fragment between the direct and predicted fits. However, the fragments counted within the area of interest, shown by the black rectangular outline, were not as close, with a direct count of 751 fragments and a predicted count of 636 fragments, yielding a difference of 115 fragments.

The total velocity of all the intersecting fragments over the entire surface from the direct fit model is 146,409 m/s and from the predicted model is 146,384 m/s, yielding a difference of 25 m/s. Like the count predictions, the total velocity within the boundary of interest was not as close, with a high difference of 3,505 m/s. However, the average values of 26 m/s and 25 m/s, for the direct fit and predicted models respectively, within the boundaries of interest, yield a difference of 1 m/s.

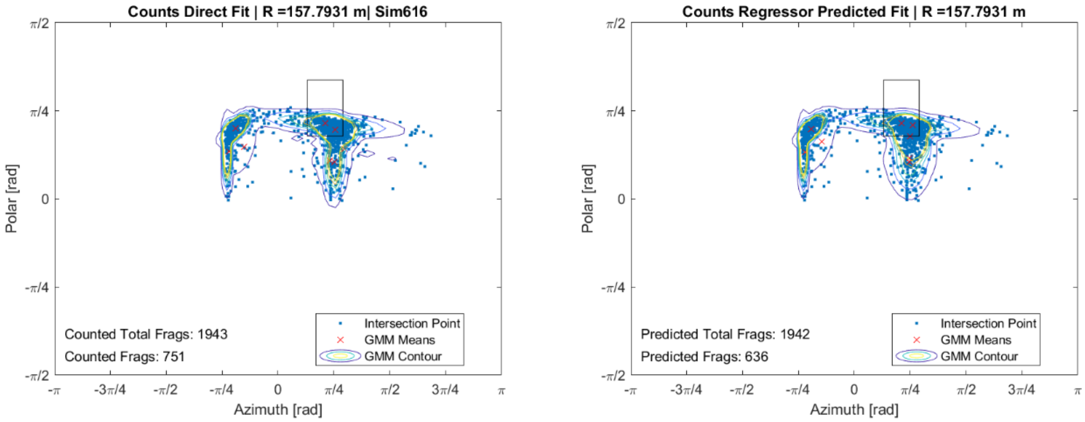


Fig. 17. Monte Carlo simulation results for count differences.

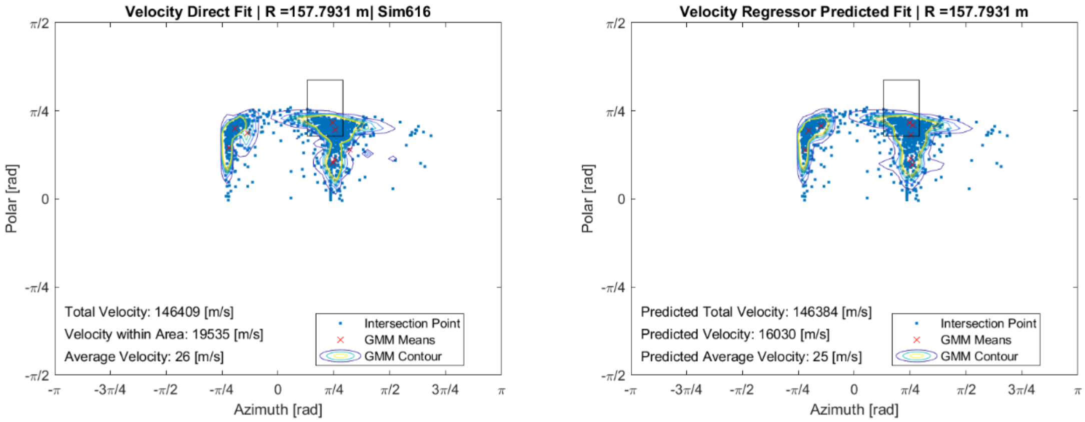
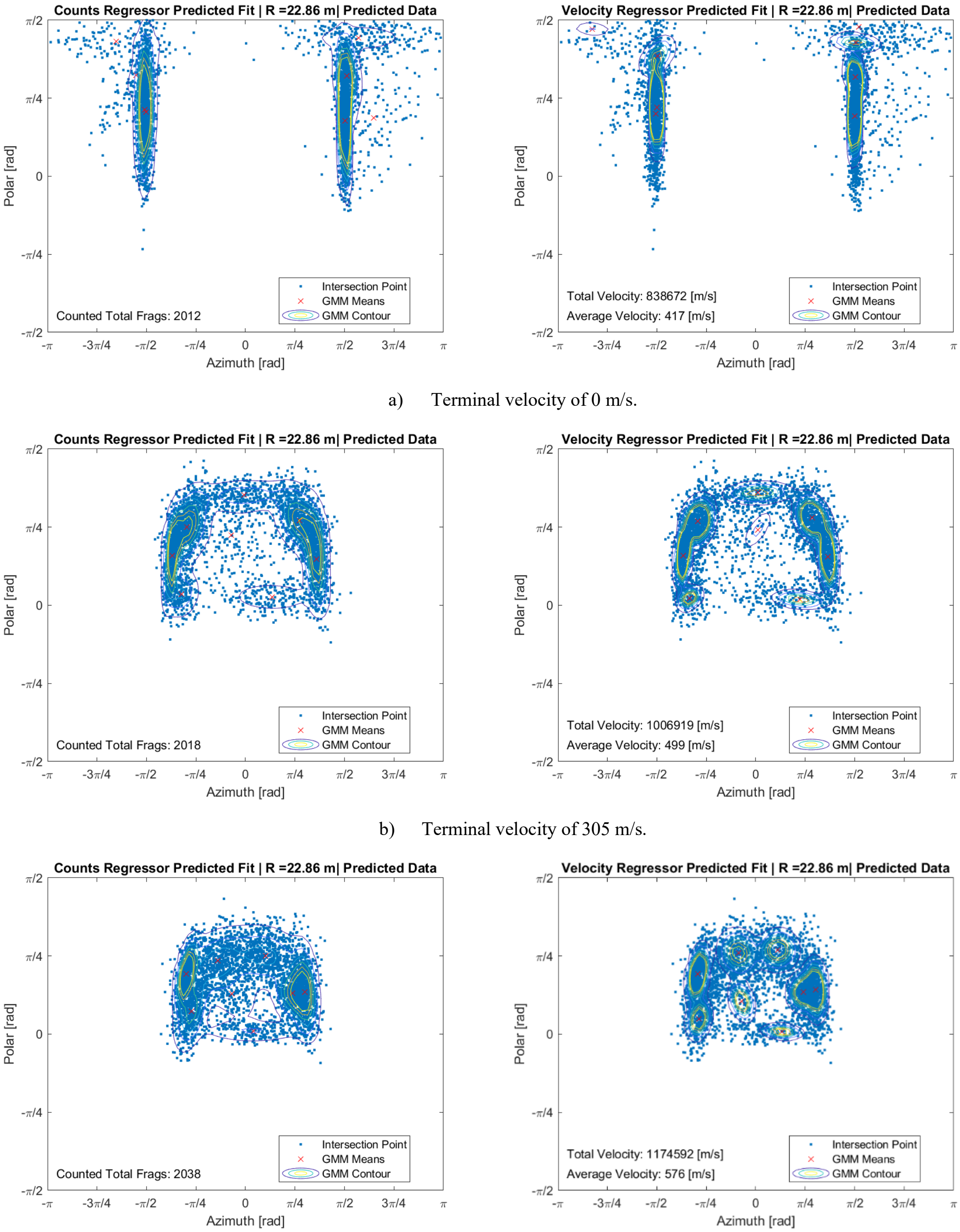


Fig. 18. Monte Carlo simulation results for velocity differences.

Visual inspection of the plots in Fig. 17 and Fig. 18, shows similar distributions between the direct fit and the predictor. However, for many of the simulation cases one can see that the total values over the entire surface of intersection are often accurate, while the values within the polar-azimuth boundary of interest are not always close. One possible explanation for the difference in values within this boundary might be that the Monte Carlo simulation is producing areas of interest with sparse data, unlike the provided example. Another likely explanation is small differences in the distributions. While the contours look relatively similar, slight differences may result in great count and velocity differences within the region of interest.

To further examine the results at various terminal conditions, as one of the desired goals is to generalize the model, the prediction tool was also examined for a variety of terminal conditions. The plots in Fig. 19 display an example of this visual comparison for varying terminal velocities (0 m/s, 305 m/s, 610 m/s, 915 m/s, 1,220 m/s, 1,525 m/s), a terminal attitude of 0° roll, pitch, and yaw, and an intersection radius of $22.86\ m$.

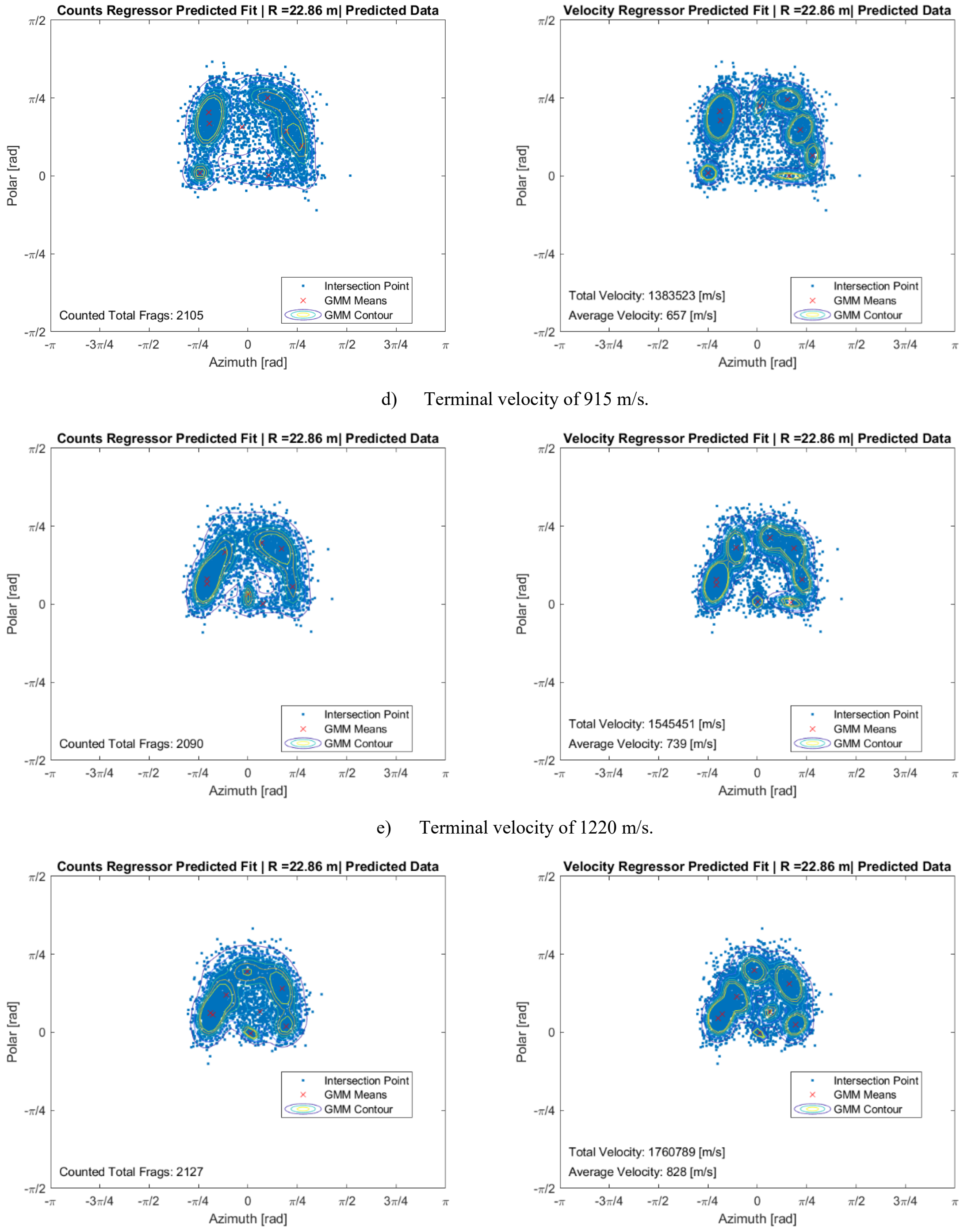


a) Terminal velocity of 0 m/s.

b) Terminal velocity of 305 m/s.

c) Terminal velocity of 610 m/s.

Fig. 19. Predicted distributions at a randomly generated radius of intersection for various denoted initial velocities.



d) Terminal velocity of 915 m/s.

e) Terminal velocity of 1220 m/s.

f) Terminal velocity of 1525 m/s.

Fig. 19. (Continued).

The trend seen is as expected. The total and average velocity increases with increasing terminal speed, while the counted total fragment predictions remain approximately the same. Additionally, the probability contours follow the trend produced by the fragment intersections, represented by the blue points on the plots. As the terminal velocity increases, the points converge to the center of the polar-azimuth plot and the distribution contours follow this course. Therefore, this proposed method is a suitable method to predict trends at various terminal conditions.

As another form of validation, the average fragment velocity was estimated using four different techniques: the static arena test, the corresponding simulation case, Gurney's equation, and the K-NN regression learning. It should be noted, for the static arena test, the initial fragment velocities were calculated using position information and interpolation, and not all 3861 fragments were collected. Therefore, the simulation velocity was used as the primary base for comparison. This was performed for a case with a 0 m/s terminal velocity and a terminal orientation of 0º roll, pitch, and yaw, displayed in Table 5.

Table 5. Comparison of the average initial fragment velocity.

Technique	Average Initial Fragment Velocity, m/s
Static Arena Test	826
Simulation Case	1002
Gurney's Equation	1036
K-NN Regression	1118

Assuming all 3861 fragments are produced, Gurney's equation and the proposed regression technique yield a percent difference of 3.4% and 12%, respectively, when compared to the average initial velocity in the corresponding simulation case. It should also be noted that assumptions were made for each technique used, therefore this test should not be used to conclude the best estimation method. However, it shows that, though the initial velocity information is not included in the training dataset, predictions can be made for inputs outside of the domain of the training set.

Using the same method as used to estimate the fragment count and fragment velocity distributions from the ball bearing training data, the fragment count, velocity, and mass distributions were estimated using the training data generated from the naturally fragmenting system. The mean difference for each characteristic, found using Monte Carlo simulations, is displayed in Table 6.

Table 6. Mean of the realistic fragmenting characteristic differences.

Characteristic	Mean Difference
Fragment Counts, Fragments	0.876
Total Velocity, m/s	5400.021
Average Velocity, m/s	711.287
Total Mass, kg	1.082
Average Mass, kg	0.185

An example of the produced distributions from the Monte Carlo simulations for the naturally fragmenting article is displayed in Fig. 20, fragment count differences, Fig. 21, fragment velocity differences using EM training data, and Fig. 22, fragment mass differences. From Fig. 22, the total mass within the boundary of interest has a difference of 5 kg between the direct fit and the predicted fit, yielding a percent difference of 20%. However, the average mass within this boundary of interest is the same for both distributions.

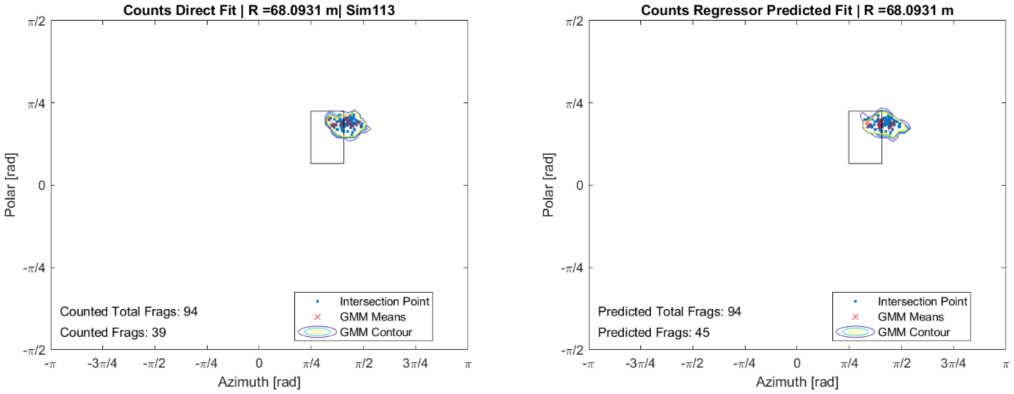


Fig. 20. Monte Carlo simulation results for count differences.

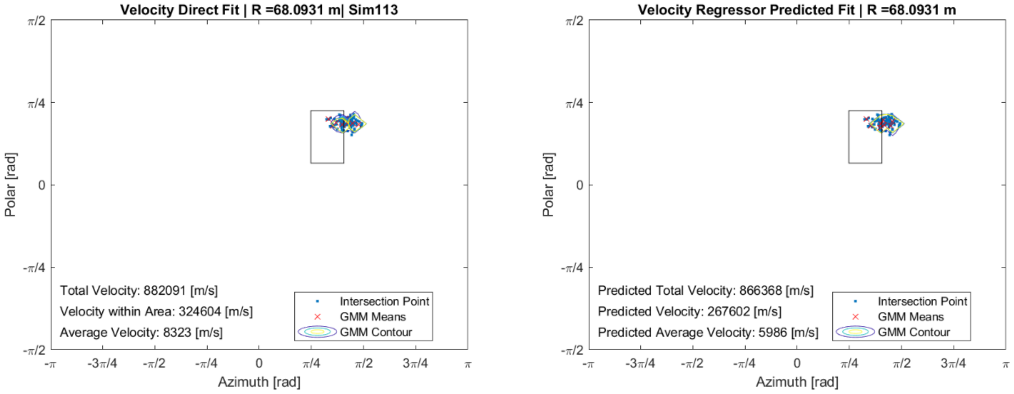


Fig. 21. Monte Carlo simulation results for velocity differences.

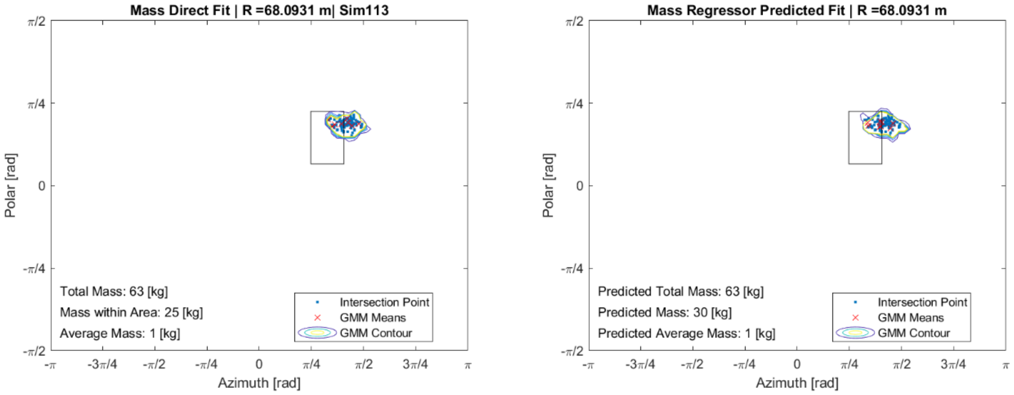


Fig. 22. Monte Carlo simulation results for mass differences.

7. Conclusions and suggestions for future work

Overall, the proposed model successfully improved count predictions from previous work and produced reasonable estimates for fragment velocities and masses. After generating training data as GMMs with and without EM, K-NN regression was compared to RFR, the previous regression method selected. The K-NN regressor with $k = 2$ was found to produce the most accurate results. Two regressors were trained. One to predict total fragment counts and another to predict total fragment velocities over an entire sphere of intersection. Integrating the predicted GMMs, the total values within a chosen boundary of interest, on a polar-azimuth coordinate system, were also calculated. This model was evaluated using Monte Carlo simulations. The same method was then repeated using natural fragmentation data to predict fragment count, velocity, and mass distributions. While the model may have some error still, it proves to make successful predictions that follow expected physical trends. This work shows that there is potential in the use of machine learning to predict fragmentation characteristics for various detonation scenarios. In the future, this work can be directly applied to space debris estimation.

Under consideration for future work are trials with different techniques to improve the model. Regression learning techniques have been the primary focus thus far, but neural networks are also under consideration. With neural networks, more complex trends might be learned to improve the model. This method could also allow the introduction of physical constraints into the training process to prevent any predictions that disobey expected behavior.

Also under consideration is the augmentation of different systems. While this research does present potential in this novel method, more realistic and diverse data should be studied as well. For example, the simulation data, though variable in impact conditions for the object, does not consider various conditions for the target, such as structure and material, which would also play a role in how the fragments propagate.

This research successfully produced and improved the distribution estimates of fragment characteristics. Therefore, there is potential for this machine learning method in fragmentation fly-out prediction, especially when only static arena test data is available.

Acknowledgements

Data is provided by the U.S. Naval Air Warfare Center Weapons Division at China Lake and high-fidelity simulations were also performed by the U.S. Naval Air Warfare Center Weapons Division. Investigations are supported through the U.S. Air Force Office of Scientific Research (award number FA9550-20-1-0200) and Embry-Riddle Aeronautical University through support for Dr. Bevilacqua during his research tenures at Eglin Air Force Base. K. E. Larsen also thanks the SMART Scholarship Program for financial support in the pursuit of her degree.

References

[1] Liou, J C. “Orbital Debris Modeling.” NASA Orbital Debris Program Office, 2012.

[2] Johnson, N L, et al. “NASA's New Breakup Model of Evolve 4.0.” *Advances in Space Research*, vol. 28, no. 9, 2001, pp. 1377–1384., doi:10.1016/s0273-1177(01)00423-9.

[3] Krisko, Paula H, and J C Liou. “NASA Long-Term Orbital Debris Modeling Comparison: Legend and Evolve.” *International Astronautical Congress of the International Astronautical Federation, the International Academy of Astronautics, and the International Institute of Space Law*, 2003, doi:10.2514/6.iac-03-iaa.5.2.03.

[4] Liou, J C, and N L Johnson. “Instability of the Present LEO Satellite Populations.” *Advances in Space Research*, vol. 41, no. 7, 2008, pp. 1046–1053., doi:10.1016/j.asr.2007.04.081.

[5] “DebriSat.” *Astromaterials Research & Exploration Science: NASA Orbital Debris Program Office*, NASA, orbitaldebris.jsc.nasa.gov/measurements/debrisat.html.

[6] Rivero, M, et al. “DebriSat Fragment Characterization System and Processing Status.” *International Astronautical Congress*.

[7] Connell, John C., et al. “Examples of Technology Transfer from the SDIO Kinetic Energy Weapon Lethality Program to Orbital Debris Modeling.” 1991, doi:10.21236/ada344572.

[8] Mikhail, Ameer G, and Alex G Kurtz. “U.S. Air Force Tests Steel Fragments Against Aluminum, Assesses Penetration of Fuel Tanks.” *Aerospace America*, 30 Nov. 2018, aerospaceamerica.aiaa.org/year-in-review/u-s-air-force-tests-steel-fragments-against-aluminum-assesses-penetration-of-fuel-tanks/.

[9] McKnight, Darren, et al. "Refined Algorithms for Structural Breakup Due to Hypervelocity Impact." *International Journal of Impact Engineering*, vol. 17, no. 4-6, 1995, pp. 547–558., doi:10.1016/0734-743x(95)99879-v.

[10] Francesconi, Alessandro, et al. "CST: A New Semi-Empirical Tool for Simulating Spacecraft Collisions in Orbit." *Acta Astronautica*, vol. 160, 2019, pp. 195–205., doi:10.1016/j.actaastro.2019.04.035.

[11] Wu D, Rosengren A. An investigation on space debris of unknown origin using proper elements and neural networks. Research Square; 2023. DOI: 10.21203/rs.3.rs-2432104/v1.

[12] N.F. Mott, Fragmentation of Shell Cases, Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences. 189 (1947) 300–308. doi:10.1098/rspa.1947.0042.

[13] E.A. Cohen, New Formulas for Predicting the Size Distribution of Warhead Fragments, Mathematical Modelling. 2 (1981) 19–32. doi:10.1016/0270-0255(81)90008-7.

[14] P. Elek, S. Jaramaz, Fragment Mass Distribution of Naturally Fragmenting Warheads, FME Transactions. (n.d.).

[15] R.W. Gurney, The Initial Velocities of Fragments From Bombs, Shell, Grenades., (1943). doi:10.21236/ada289704.

[16] V.M. Gold, E.L. Baker, J.M. Hirlinger, K.W. Ng, A Method for Predicting Fragmentation Characteristics of Natural and Performed Explosive Fragmentation Munitions, (2001). doi:10.21236/ada403989.

[17] R.E. Tipton, The CALE User's Manual, (1991).

[18] V.M. Gold, E.L. Baker, W.J. Poulos, B.E. Fuchs, PAFRAG Modeling of Explosive Fragmentation Munitions Performance, International Journal of Impact Engineering. 33 (2006) 294–304. doi:10.1016/j.ijimpeng.2006.09.032.

[19] V.M. Gold, E.L. Baker, W.J. Poulos, Modeling Fragmentation Performance of Natural and Controlled Fragmentation Munitions, (2007).

[20] V.M. Gold, Fragmentation Model for Large L/D (Length Over Diameter) Explosive Fragmentation Warheads, Defence Technology. 13 (2017) 300–309. doi:10.1016/j.dt.2017.05.007.

[21] V.M. Gold, E.L. Baker, J.M. Pincay, Computer Simulated Fragmentation Arena Test for Assessing Lethality and Safety Separation Distances of Explosive Fragmentation Ammunitions, Computational Ballistics III. (2007). doi:10.2495/cbal070171.

[22] R.N. Tiile, Artificial Neural Network Approach to Predict Blast-Induced Ground Vibration, Airblast and Rock Fragmentation, thesis, 2016.

[23] I. Enayatollahi, A. Aghajani Bazzazi, A. Asadi, Comparison Between Neural Networks and Multiple Regression Analysis to Predict Rock Fragmentation in Open-Pit Mines, Rock Mechanics and Rock Engineering. 47 (2013) 799–807. doi:10.1007/s00603-013-0415-6.

[24] R.F. Hay, G.M. Gibson, M.P. Lee, M.J. Padgett, D.B. Phillips, Four-Directional Stereo-Microscopy for 3D Particle Tracking with Real-Time Error Evaluation, Optics Express. 22 (2014) 18662. doi:10.1364/oe.22.018662.

[25] J. He, A. Spencer, E. Chu, OpenCV and TYZX : Video Surveillance for Tracking., (2008). doi:10.2172/942060.

[26] S. King, Camera System Captures, Analyzes Munition Detonation Data, Air Force Materiel Command. (n.d.). https://www.afmc.af.mil/News/Article-Display/Article/2975486/camera-system-captures-analyzes-munition-detonation-data/.

[27] O.S. Mulekar, R. Bevilacqua, E.L. Jerome, T.J. Hatch-Aguilar, Transfer function to predict warhead fragmentation in-flight behavior from Static Data, AIAA Journal. 59 (2021) 4777–4793. doi:10.2514/1.j060226.

[28] V. Gerbaud, Y. Demirel, Probabilistic Approach in Thermodynamics, in: Nonequilibrium Thermodynamics: Transport and Rate Processes in Physical, Chemical and Biological Systems, ELSEVIER Science LTD, 2019: pp. 711–791.

[29] F. Gensdarmes, Methods of Detection and Characterization, in: P.I. Dolez (Ed.), Nanoengineering: Global Approaches to Health and Safety Issues, Elsevier Science, San Diego, CA, 2015: pp. 55–84.

[30] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, et al., Scikit-learn: Machine Learning in Python, Journal of Machine Learning Research. 12 (2011) 2825–2830.

[31] P. Eslambolchilar, A. Komninos, M. Dunlop, Machine Learning Basics, in: Intelligent Computing for Interactive System Design: Statistics, Digital Signal Processing, and Machine Learning in Practice, Association for computing machinery, New-York, n.d.: pp. 143–193.

[32] J. McGonagle, G. Pilling, A. Dobre, V. Tembo, A. Kurmukov, A. Chumbley, et al., Gaussian Mixture Model, Brilliant Math & Science Wiki. (n.d.). https://brilliant.org/wiki/gaussian-mixture-model/#:~:text=A%20Gaussian%20mixture%20of%20three,subpopulations%20within%20an%20overall%20population.

[33] I.D. Gebru, X. Alameda-Pineda, F. Forbes, R. Horaud, EM Algorithms for Weighted-Data Clustering with Application to Audio-Visual Scene Analysis, IEEE Transactions on Pattern Analysis and Machine Intelligence. 38 (2016) 2402–2415. doi:10.1109/tpami.2016.2522425.

[34] S. Theodoridis, K. Koutroumbas, Pattern Recognition, 4th ed., Academic Press, Inc., 2008.

[35] O. Kramer, Unsupervised K-Nearest Neighbor Regression, (2011).

[36] Distance Weighted K-NN Algorithm, (n.d.). http://www.data-machine.net/nmtutorial/distanceweightedknnalgorithm.htm.

[37] K.S. Kannan, K. Manoj, Outlier Detection in Multivariate Data, Applied Mathematical Sciences. 9 (2015) 2317–2324. doi:10.12988/ams.2015.53213.

[38] D.P. Kroese, T. Brereton, T. Taimre, Z.I. Botev, Why the Monte Carlo Method is so Important Today, WIREs Computational Statistics. 6 (2014) 386–392. doi:10.1002/wics.1314.