BULLETIN OF THE POLISH ACADEMY OF SCIENCES
TECHNICAL SCIENCES, Vol. 74(3), 2026, Article number: e158297
DOI: 10.24425/bpasts.2026.158297

Approximation of two-dimensional ballistic limit curves in the range of material thickness with stochastic physical-based oversampling

Tomasz HACHAJ[1]* **and Teresa FRĄŚ**[2]

[1] Faculty of Electrical Engineering, Automatics, Computer Science and Biomedical Engineering, AGH University of Krakow, Al. Mickiewicza 30, Krakow 30-059, Poland
[2] French-German Research Institute of Saint-Louis (ISL), 5 rue du Général Cassagnou, 68301, France

Abstract. In this paper, we propose a method for approximating ballistic curves and determining ballistic limit velocity based on a data-driven approach as a function of two variables: thickness and initial velocity, allowing for the approximation of these parameters for material thicknesses that were not present in the training set. This differs from previous work in this area, where a random division was made between the training and validation sets, which did not guarantee separability in terms of material thickness between the training and validation sets. To prove the effectiveness of this approach, we performed leave-one-out cross-validation. Our method was trained on ballistic experimental data, which was extended using the finite element method. We also proposed a new method of data oversampling based on fitted ballistic curves estimated using the Recht-Ipson method. Oversampling involves the use of stochastic sampling in which the cumulative distribution function is a mixture of uniform sampling and the first and second derivatives derived from ballistic limit curves. We have evaluated several deep neural network architectures. Our experiments have shown that it is possible not only to approximate the shape of the curve but also to accurately predict the ballistic limit velocity for material thicknesses not present in the dataset. The inclusion of information about the first and second derivatives in the stochastic oversampling process allowed for a significant increase in prediction accuracy over uniform sampling.

Keywords: machine learning; terminal ballistic; stochastic; physical-based; oversampling.

1. INTRODUCTION

Research and modeling of terminal ballistics issues is a challenge for scientists and engineers. This is due to the speed of projectiles moving toward the target, the very short duration of the collision with the target, and the complexity of the projectile's penetration process, followed by the fragmentation of both objects. In practice, achieving repeatability of the initial conditions of the phenomena under study and standardizing observations is very costly and requires specialized equipment and specialized research staff. For these reasons, in some cases, in addition to experimental data, ballistic research uses simulations, such as the finite elements method (FEM), which complement experimental measurements. The usefulness of data-driven modeling using machine learning (ML) techniques is increasingly being analyzed for specific terminal ballistics issues.

1.1. State-of-the-art

Some recent studies propose ML models designed to approximate the outcome parameters of an experiment using selected initial parameters and information about the materials under investigation. For example, in [1], a deep neural network (DNN) is proposed to assist in identifying blast-loading scenarios that predict out-of-plane displacement. The authors tested five relatively simple feedforward networks with between two and eight hidden layers, differing in the number of neurons with Sigmoid or Tanh activation functions. The authors also used the characteristic [2] explaining the impact of input parameters on network performance. The work [3] used several popular ML algorithms, such as extreme gradient boosting (XGBoost), artificial neural network (ANN), support vector regression (SVR), and Gaussian process regression (GP) were applied to two typical terminal ballistics problems: predicting the ballistic limit V50 [4] of monolithic metal armor under the impact of small- and medium-caliber projectiles and fragments, and predicting the depth to which a projectile penetrates a target of semi-infinite thickness. The authors of [5] created a model based on a generative adversarial network (GAN), trained using data from ballistic testing, which allows for generating ballistic samples instead of performing additional destructive experiments. In [6], an ANN was used to approximate the relation between target thickness, impact angle, and projectile nose shape on the ballistic limit velocity. Another promising application of ML methods in terminal ballistics is the use of physics-informed neural networks (PINNs) [7, 8] proposed in [9], in which the authors incorporate physics knowledge in the form of existing ballistic limit equations into the network training process to solve the classification problem of projectile penetration through material.

*e-mail: thachaj@agh.edu.pl

Manuscript submitted 2025-11-13, revised 2026-02-10, initially accepted for publication 2026-02-22, published in May 2026.

Knowledge about the physical properties of the modeled problem can be applied in ML not only in the optimization process but also in the process of generating additional samples (oversampling), which can improve the performance of the model in both classification and regression problems [10, 11]. Suppose we want to use the samples generated in this way to train a data model that will allow us to extend the known model with additional input parameters. In that case, the sampling must not only satisfy the fundamental Nyquist-Shannon sampling theorem that a band-limited signal can be perfectly reconstructed from its samples if the sampling rate is at least twice the highest frequency in the signal [12], but it must also address a crucial gap between the approximation theory of DNNs trained with stochastic gradient descent and their practical performance on, for example, smooth functions [13, 14]. Although deep learning has the potential to achieve significant performance gains, it is poorly understood from a theoretical perspective, especially regarding sample complexity – the amount of training data needed to learn good DNN approximations for specific classes of functions [15]. The process of properly preparing samples for machine learning training remains an open issue. A naive approach to such problems often leads to sample waste and inefficient approximation schemes. For example, uniform sampling can result in significant sample waste in approximation problems [16].

1.2. Novelty of this paper

Based on the state-of-the-art, the application of ML methods in terminal ballistics is a promising research area with great potential for practical applications, particularly in expanding knowledge about modeled phenomena through data-driven learning. In this paper, we propose a method for approximating ballistic curves and determining ballistic limit velocity based on data (data-driven approach) as a function of two variables: thickness and initial velocity, allowing for the approximation of these parameters for material thicknesses that were not present in the training set. This differs from previous work in this area, where a random division was made between the training and validation sets, which did not guarantee separability in terms of material thickness between the training and validation sets. To prove the effectiveness of this approach, we performed leave-one-out cross-validation. Our method was trained on ballistic experimental data extended with FEM data. We also proposed a new method of data oversampling based on fitted ballistic curves estimated using the Recht-Ipson (R-I) method [17]. Oversampling involves the use of stochastic sampling in which the cumulative distribution function is a mixture of uniform sampling and the first and second derivatives derived from ballistic limit curves (physically-based oversampling). Our goal is not to replace FEM simulations with machine learning models, but only to demonstrate that, by using knowledge about material properties (in our case, analytical R-I formula) fitted by a relatively small number of samples and appropriately constructed stochastic sampling, it is possible to model ballistic limit curves effectively.

2. MATERIAL AND METHODS

In this section, we will discuss ballistic experiments that result in ballistic limit curves and the dataset created from these experiments. Next, we will introduce the probabilistic sampling method, which uses analytical formulas describing the phenomena under study. Next, we will use approximation models based on deep neural networks. This approach will enable us to define and train machine learning models that expand our knowledge of the analyzed phenomenon with additional parameters for which no analytical formulas exist.

2.1. Problem formulation and the dataset

The data set we used in our work consisted of data from an experimental investigation aimed at determining the ballistic limit curves for S355 steel plates subjected to impacts from small-calibre 7.62×51 mm P80 (0.308 Win) armor-piercing rounds. The ballistic impact experiments concerned two steel target thicknesses, 8 mm and 16 mm. The tests were performed at an instrumented terminal ballistic laboratory, which allowed for in-situ measurements and observations. The initial impact velocity of the projectiles was measured by the optical light barrier. Prior to the impact, the pitch and yaw angles of the projectiles were measured to ensure that the analyzed shots were not affected by too high an initial yaw impact angle. Each shot was captured from the side using flash X-ray imaging to observe the projectile behaviour and to calculate its residual velocity. In all cases which resulted in the perforation of the S355 target plates (of thicknesses 8 mm and 16 mm), the armour-piercing rounds were not damaged or fragmented (Fig. 1). Due to the shots with varied velocities, the ballistic limit curves and the ballistic limit velocities for the two test configurations were determined (Fig. 3a).

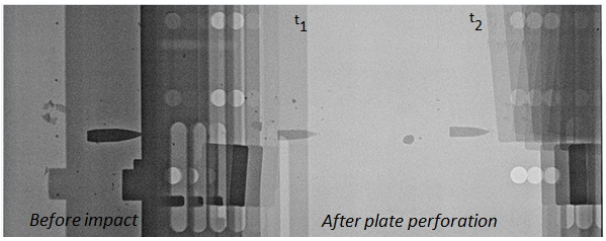


Fig. 1. An example of an X-ray image taken during a shot at an 8 mm thick plate, with an initial velocity of 473 m/s

Ballistic curves show the relationship between the velocity of a projectile immediately before impact with a material (initial velocity v_i) and the velocity of the projectile after impact with the material and possible penetration (residual velocity v_r). If the projectile fails to penetrate the material, we assume that its v_r is 0. An important parameter of the material is the so-called ballistic limit velocity v_{bl}. It is assumed that this is the lowest velocity below which projectiles are stopped by the material and above which projectiles penetrate the material. In practice, v_{bl} is determined statistically. The shape of ballistic curves is often approximated using the Recht-Ipson model [17], which defines

the following equation:

$$v_r \approx RI(v_i, a, p) = \begin{cases} a \cdot (v_i^p - v_{bl}^p)^{\frac{1}{p}} & \text{when } v_i \geq v_{bl}, \\ 0 & \text{when } v_i < v_{bl}, \end{cases} \quad (1)$$

where v_i – initial velocity, v_r – residual velocity, v_{bl} – the sought ballistic limit velocity, a – parameter calculated considering mass of the projectile and its possible plugs and fragments, a, p - parameter to be fitted from experimental data. Parameter fitting is often performed by a numeric solver, for example, nonlinear least squares with bounds [18] or generalized reduced gradient method [19]. Note that the function (1) is differentiable in the interval $(-\infty, \infty)$, but does not have a second derivative at the point v_{bl}. The initial form of equation (1) proposed in [20] was derived by analyzing the energy balance during the process of a projectile penetrating a thin plate, and it has physical justification.

In our investigation, steel samples with thicknesses of 4 mm, 6 mm, 8 mm, 10 mm, 12 mm, 14 mm, 16 mm, and 18 mm were analyzed. For 8 mm and 16 mm, an experiment was conducted to determine the relationship between initial and residual velocities, aiming also to calibrate the numerical model. For the remaining thicknesses, a simulation was performed using the finite element method (FEM).

The goal of the numerical simulation using the finite element method (FEM) in the Lagrangian solver of Impetus Afea was to expand the experimental dataset. The Impetus software automatically generated the meshes for the components of the impact configurations, including the 7.62 P80-caliber projectile and the steel target with various thicknesses (Fig. 2). The software automatically defined the contact, allowing both configuration components to fail due to a contact. The numerical model did not account for pitch and yaw angles upon the impacts. The details of modeling approaches may be found in the Impetus manual [21] and also in, e.g., [22] or [23]. The models defining the behavior of the materials used to build the projectile were proposed by the Impetus library and applied to all components, including both the target and the projectile. The flow and fracture models of the steel S355, as well as those of the projectile brass jacket and its lead cup, are implemented using generic functions proposed by the software developers. To model the flow and fracture behavior of the projectile core, the classical Johnson-Cook modeling approach has been used, [24] and [25]. This well-known model contains the three sets of brackets that represent the strain hardening, strain rate hardening and thermal softening to describe the flow of a ductile metallic structure. Whereas, in the Johnson and Cook fracture strain model, the damage evolution takes into account the influence of stress triaxiality, strain rate, and temperature on the fracture strain, e.g., [26–28]. The parameters of the model of the high-strength steel of the projectile core, originated from [29] and [30].

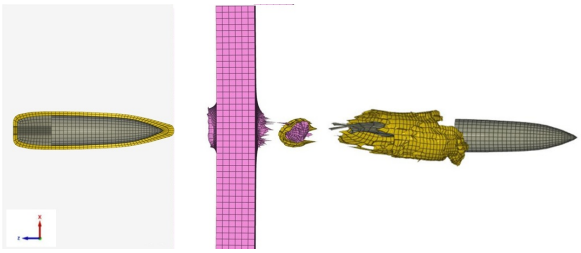


Fig. 2. Numerical configuration: cross-section of the projectile before impact and after the plate perforation

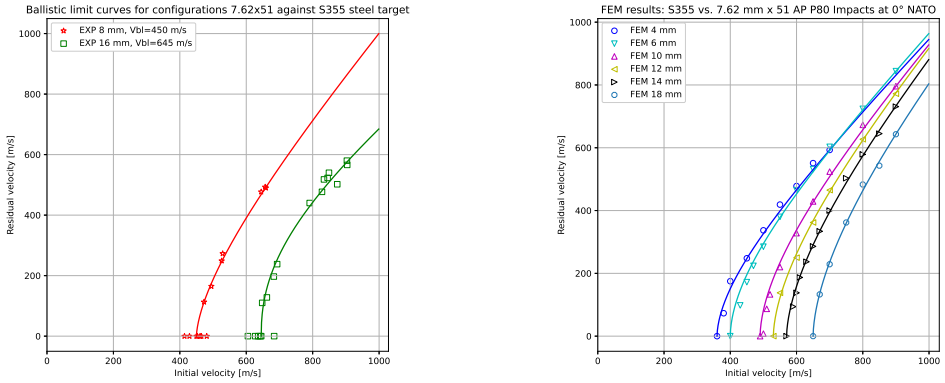
Figure 3 shows the entire set of measurement points with the corresponding fitted ballistic curves. The data from the experiments (blue circles and black triangles) are marked in Fig. 3 with the abbreviation EXP, and the remaining data from the simulations (numerical data) are marked with the symbol FEM.

The use of FEM to supplement data in ballistic experiments is well justified. Conducting ballistic experiments is costly because it requires specialized equipment and skilled personnel. However, it should be remembered that the results obtained through simulation cannot replace actual experimental data. Furthermore, FEM calculations of complex ballistic models require significant computing resources. Even in dedicated computer centers, performing a full simulation for a single material sample with given physical parameters is a lengthy process. For this reason, there is a justified need to create numerical models that generalize known relationships, such as (1), so that the behavior of materials for other sets of input parameters can be modeled accurately and quickly. For this reason, the possibilities of applying various machine learning methods to approximating such relationships are increasingly being explored. A significant problem with using machine learning for ballistics issues is the limited data available from experiments and the necessity of performing time-consuming simulations to generate numerical data. This problem can be solved by using stochastic oversampling, which utilizes known physical parameters of materials to create new samples for ML training, ensuring they retain the required constraints. In the case of experiments involving determining ballistic limit curves, we can accurately model dependencies between v_i and v_r material with specified thicknesses using (1), which we then want to generalize to material thicknesses that were not the subject of the experiments. In other words, we want to extend (1) with an additional parameter related to thickness:

$$\widehat{v_r}(v_i, t, \Pi) : \min_{\Pi} ERR(\widehat{v_r}, v_r), \quad (2)$$

where t is a material thickness, Π are the parameters of the ML model that minimize the error ERR between the data derived from the model predictions $\widehat{v_r}$ and the data from the training set v_r.

Our goal is not to replace FEM simulations with machine learning models, but only to demonstrate that, by using knowledge about material properties (in our case, analytical R-I formulas (1)) fitted by a relatively small number of samples and appropriately constructed stochastic sampling, it is possible to model ballistic limit curves effectively. Since ML approximations are usually very fast, it will be possible to use them to model ballistic limit curves across the entire thickness range of a given material $(t_{\min}, t_{\max})$, where these are, respectively, the smallest and largest thicknesses of materials for which we have



(a) Experimental results

(b) Simulation results with finitie element method

Fig. 3. The entire set of measurement points in our dataset with the corresponding fitted ballistic curves. The data from the experiments (blue circles and black triangles) are marked in Fig. 3 with the abbreviation EXP, and the remaining data from the simulations (numerical data) are marked with the symbol FEM

experimental or numerical data and for which we have calculated ballistic limit curves. This assumption excludes extrapolation for values smaller than $t_{\min}$ and larger than $t_{\max}$.

2.2. Calculation of cumulative distribution function for probabilistic sampling

To train a machine learning model, a statistically representative and sufficiently sized data sample must be provided in the training set, allowing observation of significant nuances in the variability of the modeled function. It is obvious that if the modeled function has global or local extremes that need to be mapped for the machine learning model to be useful in real-world applications, the training data set must be prepared with enough samples for this mapping. While the issue of a "sufficiently large data set" is difficult to define precisely, it is possible to define data sampling in such a way that it includes those areas of the modeled function where there is proportionally high variability in its course. The simplest way to collect data is uniform sampling, in which each value in a given interval $[b_1, b_2]$ has the same probability of being selected. However, such sampling does not take into account the fact that in some areas the function may be increasing or decreasing. This information is contained in the absolute value of the first derivative. Another important piece of information is acceleration, understood as a change in the increment of the variable. Information about acceleration is contained in the second derivative of the function. Sampling a random variable should therefore consider both uniform sampling and information about the first and second derivatives of the function. This approach ensures that the diversity of the data is fully reflected, enabling the machine learning method to learn such variability correctly. We will now define the cumulative distribution function (CDF), which will take into account the three values mentioned above.

Let us assume that $f(x)$ is a function that has first and second derivatives in the interval $[b_1, b_2]$. Using the function $f(x)$, let us define the function $g(x)$, which is the weighted sum of the constant term and the absolute values of the first and second derivatives of the function $f(x)$.

$$g(x)_{[b_1,b_2]} = c_1 + c_2 \cdot \left|\frac{df(x)}{dx}\right| + c_3 \cdot \left|\frac{d^2 f(x)}{dx^2}\right|, \tag{3}$$

where $x \in [a,b] \wedge \ c_1 + c_2 + c_3 = 1 \wedge \ c_1, c_2, c_3 \in [0,1]$, $|x|$ is absolute value of x.

Next, let us define a function:

$$F_x(v) = P(x \le v) = \frac{\int_{b_1}^{v} g(x)_{[b_1,b_2]}\,\mathrm{d}x}{\int_{b_1}^{b_2} g(x)_{[b_1,b_2]}\,\mathrm{d}x}. \tag{4}$$

Function (4) is continuous monotone nondecreasing function in $[b_1, b_2]$ which satisfies conditions:

$$F_x(b_1) = \frac{\int_{b_1}^{b_1} g(x)_{[b_1,b_2]}\,\mathrm{d}x}{\int_{b_1}^{b_2} g(x)_{[b_1,b_2]}\,\mathrm{d}x} = 0 \tag{5}$$

and:

$$F_x(b_2) = \frac{\int_{b_1}^{b_2} g(x)_{[b_1,b_2]}\,\mathrm{d}x}{\int_{b_1}^{b_2} g(x)_{[b_1,b_2]}\,\mathrm{d}x} = 1. \tag{6}$$

Based on the above calculations, the function (4) in the interval $[b_1, b_2]$ can be used to generate random samples of data with a distribution in which the parameters $[c_1, c_2, c_3]$ define the proportion of uniform distribution and distribution proportional to the first and second derivatives of the function $f(x)$ in a random sample. We can use CDF directly to draw a random sample of data using, for example, inverse transform even

from a non-Gaussian distribution [31] or with the help of other methods [32]. Equation (4) can be used if we substitute (1) for $f(x)$, but we must exclude the point v_{bl} where the second derivative (1) is discontinuous. For this reason, we can calculate the CDF for v_r in the interval $[v_{bl}+\epsilon, \inf)$, where ϵ is a small number greater than zero. To use the CDF calculated in this way to generate a sample for machine learning training, we obviously limit the domain to the interval in which we perform the approximation, so b_2 is a finite value.

In Fig. 4, we present the first and second derivatives for the example function 1 from our dataset (thickness = 16 mm) and the CDF values calculated according to the formula (4) for different values of the parameters $[c_1, c_2, c_3]$ in range $[v_b l + 0.05, v_b l + 200]$. We chose this range to ensure that both the characteristics of the curves near v_{bl} and the upward trends with increasing arguments are visible.

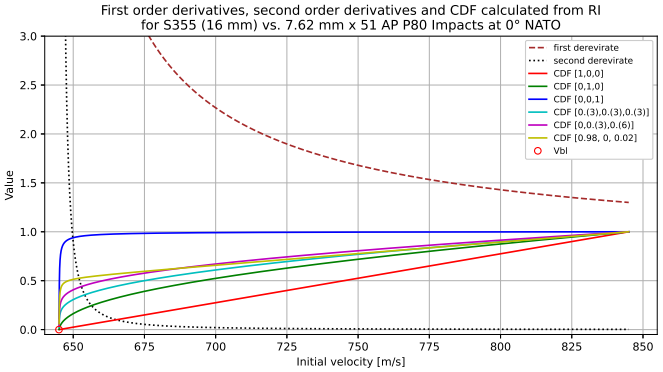


Fig. 4. The first and second derivatives for the example function (1) from our dataset (thickness = 16 mm) and the CDF values calculated according to the formula (4) for different values of the parameters $[c_1, c_2, c_3]$ in range $[v_b l + 0.05, v_b l + 200]$

We will now discuss the plots of the values considered in the examined interval. The absolute value of the first derivative represented by the brown dashed line reaches its maximum value near v_{bl} where the ballistic curve has the greatest increase. Then, because the rise in the value of the function is close to linear, the value decreases exponentially, approaching the value 1.25. The absolute value of the second derivative, representing the change in the value of the first derivative, represented by the dotted black line, reaches a value close to v_{bl} and has a greater value than the modulus of the first derivative. Since the value of the first derivative of the ballistic curve in the further considered interval is almost linear, the second derivative in that area is close to zero.

The CDF calculated using formula (4) if $c_1 = 1$ and the remaining coefficients $c_2 = c_3 = 0$ has a uniform distribution because the only nonzero component is a constant value. For this reason, the CDF is a straight line segment at point $(b_1, 0)$ and ends at point $(b_2, 1)$. In Fig. 4, it is represented by a red line.

If $c_2 = 1$ and the other coefficients $c_1 = c_3 = 0$, the CDF increases rapidly at the beginning of the interval under consideration and then approaches $(b_1, 0)$ almost linearly. In Fig. 4, it is represented by a green curve.

If $c_3 = 1$ and the other coefficients $c_1 = c_2 = 0$, the CDF takes a value close to 1 almost at the beginning of the interval under consideration and then asymptotically approaches $(b_1, 0)$ with an increment close to zero. Both of the above cases correspond perfectly to the graphs of the absolute values of the first and second derivatives of the CDF. In Fig. 4, it is represented by a blue curve.

The remaining values shown in Fig. 4 are scaled linear combinations of the previously discussed cases. When all coefficients are equal to $c_1 = c_2 = c_3 = 0.(3)$, the resulting CDF starts with a higher initial value than the first derivative and has a higher value over the entire interval under consideration. This is due to the significant contribution of the second derivative to the initial sum and the insufficient contribution of the linear uniform distribution. In Fig. 4, this case is represented by the cyan curve.

If we mix the first and second derivatives in (4) in proportions $0.(3)$ and $0.(6)$, omitting the uniform factor, we obtain a graph whose initial values are greater than the first derivative and $c_1 = c_2 = c_3 = 0.(3)$ and which grows faster than linearly. In Fig. 4, the magenta curve represents this case.

The last case considered is a mix consisting of a uniform distribution and the second derivative, in proportions of 0.98 and 0.02, respectively. The graph initially has a value greater than $[0, 0.(3), 0.(6)]$ and then grows linearly and slightly slower than the magenta curve. In Fig. 4, this case is represented by the yellow curve.

As can be easily seen, manipulating the three coefficients $[c_1, c_2, c_3]$ allows the shape of the CDF to be modified. It enables the characteristics of the curve to be considered to varying degrees in the sampling process.

Figure 5 shows example results of sampling using a CDF constructed based on (4) for different configurations of $[c_1, c_2, c_3]$

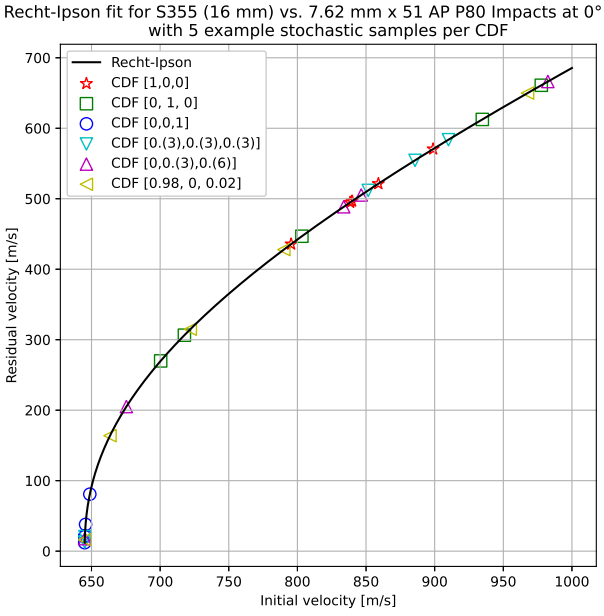


Fig. 5. Example results of sampling using a CDF constructed based on (4) for different configurations of $[c_1, c_2, c_3]$ for the same seed value used in generating pseudo-random numbers. For each CDF configuration, five samples were randomly selected

for the same seed value used in generating pseudo-random numbers. For each CDF configuration, five samples were randomly selected. When the CDF is uniform, i.e., $[1,0,0]$, the samples are evenly distributed as the initial velocity increases (see red stars). A higher value of the parameter c_2, responsible for the contribution of the first derivative to the CDF mixture, results in a larger number of samples in the initial areas of the curve, which are characterized by a higher value of the first derivative than is the case with an increase in initial velocity. This can be seen in the distribution of samples for the green squares. The contribution of c_2 to the mixture CDF increases the number of samples at the very beginning of the curve, in the immediate vicinity of v_{bl} (see blue circles). The combination of different parameters allows for a balance between these three properties provided by (4), for example, as is the case for $[0.98,0,0.02]$ (yellow triangles) and for $[0.(3),0.(3),0.(3)]$ (cyan triangles) and $[0,0.(3),0.(6)]$ (magenta triangles), where the sampled points are present both in the immediate vicinity of v_{bl} and at successive points on the ballistic curve.

Selecting the appropriate size of the random sample for training the approximation algorithm is one of the key issues. The sample should be large enough for the algorithm to adapt to the data, but not too large to prevent overfitting. No universal method has yet been proposed to determine the optimal sample size. Still, research [33, 34] suggests that for slightly simpler cases where regression was used, the minimum sample size was around 25. Since the models we used contain between 10^3 and $4.5 \cdot 10^4$ parameters, we decided to use a random sample twice as large, i.e., 50.

2.3. Approximation model

In many cases, neural networks are useful nonlinear approximators that, with a properly prepared training dataset, allow for the generalization of approximation results to data on which the network was not trained. In our study, we used a deep neural network with an architecture similar to that presented in [3]. At the input, each network (input layer) accepted the initial velocity v_i and material thickness t, and at the output, it returned the predicted residual velocity $\widehat{v_r}$. We tested the performance of six DNNs with pyramid topologies, in which each successive layer contained fewer neurons than the previous one. Each level of the pyramid consisted of three consecutive layers: a dense (fully connected) layer, followed by batch normalization (BN) and dropout. We used Scaled Exponential Linear Unit (SELU) as the activation function in the fully connected layer:

$$SELU(x) = \begin{cases} \alpha \cdot \beta \cdot (e^x - 1) & \text{when } x \le 0, \\ \beta \cdot x & \text{when } x > 0, \end{cases} \tag{7}$$

where $\alpha = 1.67326324$ and $\beta = 1.05070098$.

The role of batch normalization and dropout layers is to perform regularization, ensuring the network is not overfit and it can generalize results to unknown input parameters, particularly thicknesses that did not occur in the training set. For this reason, the dropout rate was set to a relatively high value of 0.1. The output layer of the network consisted of a single neuron with a rectified linear unit (ReLU) activation function:

$$ReLu(x) = \begin{cases} 0 & \text{when } x \le 0, \\ x & \text{when } x > 0. \end{cases} \tag{8}$$

Table 1 presents descriptions of the topologies of neural networks used in our research. All networks have the same two-dimensional input $[v_i,t]$. Each network has hidden layers following the same topology: a fully connected layer followed by batch normalization and dropout. However, the number of hidden layers and neurons in them differs. For example, the first layer has 256 neurons in a dense layer with SELU activation (7), followed by BN and dropout with a parameter of 0.1. Next, there are 128 neurons in a hidden layer with SELU activation, followed by BN and dropout with a parameter of 0.1, etc. All networks have the same fully connected output layer with ReLu activation (8) that returns predicted residual velocity $\widehat{v_r}$

Our networks, therefore, approximate the generalization of formula (1), which is extended by a parameter related to the thickness of the analyzed material. It is important to note that this approximation does not have a ready-made analytical formula. Thanks to this, the approximation we have obtained expands our knowledge of the analyzed phenomenon and is not merely a simple replacement of the analytical formula with machine learning.

Table 1
Descriptions of the topologies of neural networks used in our research

ID	Input	Dense # neurons	Hidden layers topology	Output	# param.
1	$[v_i,t]$	[256, 128, 64, 32, 16]	$Dense(SELU) \rightarrow BN \rightarrow Dropout(0.1)$	$Dense(ReLu) \rightarrow \widehat{v_r}$	45 541
2	$[v_i,t]$	[128, 64, 32, 16, 8]	$Dense(SELU) \rightarrow BN \rightarrow Dropout(0.1)$	$Dense(ReLu) \rightarrow \widehat{v_r}$	11 893
3	$[v_i,t]$	[128, 64, 32, 16]	$Dense(SELU) \rightarrow BN \rightarrow Dropout(0.1)$	$Dense(ReLu) \rightarrow \widehat{v_r}$	11 749
4	$[v_i,t]$	[64, 32, 16, 8]	$Dense(SELU) \rightarrow BN \rightarrow Dropout(0.1)$	$Dense(ReLu) \rightarrow \widehat{v_r}$	3 189
5	$[v_i,t]$	[64, 32, 16]	$Dense(SELU) \rightarrow BN \rightarrow Dropout(0.1)$	$Dense(ReLu) \rightarrow \widehat{v_r}$	3 045
6	$[v_i,t]$	[32, 16, 8]	$Dense(SELU) \rightarrow BN \rightarrow Dropout(0.1)$	$Dense(ReLu) \rightarrow \widehat{v_r}$	885

2.4. Ballistic curve fitting and estimation of ballistic limit velocity

The optimization of the parameters of the models described in Section 2.3 was performed using the stochastic gradient descent (SGD) algorithm [35], and the minimized function was the mean square error (MSE).

$$MSE(X,\widehat{X}) = \frac{1}{n}\cdot\sum_{i=1}^{n}(x_i-\widehat{x_i}), \tag{9}$$

where X is actual value, $\widehat{X}$ is a predicted value.

The MSE metric is absolute, so to further evaluate the effectiveness of the ballistic curve fit by the neural network, we will use two other metrics. The first is the coefficient of determination, which indicates the proportion of the variance in a dependent variable that is predictable from one or more independent variables (R^2):

$$R^2(X,\widehat{X}) = 1-\frac{\sum_{i=1}^{n}(x_i-\widehat{x_i})^2}{\sum_{i=1}^{n}(x_i-\overline{x})^2}, \tag{10}$$

where $\overline{x}$ is a mean value.

We also used symmetric mean absolute error (SMAE):

$$SMAE(X,\widehat{X}) = \frac{1}{n}\cdot\sum_{i=1}^{n}\frac{|x_i-\widehat{x_i}|}{\frac{|x_i|+|\widehat{x_i}|}{2}}. \tag{11}$$

To determine the prediction of the ballistic limit velocity $\widehat{v_{bl}}$ in a given interval $[b_1,b_2]$, we iterate successive values of the residual velocity prediction $\widehat{v_r(v)}$ until $\Delta\widehat{v_r(v)} > \epsilon_{v_r}$ and $\frac{d\widehat{v_r(v)}}{dv} > 0$ where ϵ_{v_r} is a threshold value. In other words, the searched value v_{bl} will be the first value v in the considered interval for which the relative change in velocity $\Delta\widehat{v_r(v)}$ will be above ϵ_{v_r} and for which the first derivative will be positive (the function $\widehat{v_r(v)}$ will be locally increasing).

In order to evaluate the precision of v_{bl} approximation we utilized absolute error (AE):

$$AE(x,\widehat{x}) = |x-\widehat{x}| \tag{12}$$

and relative error (RE):

$$RE(x,\widehat{x}) = \left|\frac{x-\widehat{x}}{x_i}\right|. \tag{13}$$

3. RESULTS

We have implemented our approach presented in Section 2 in Python 3.10 using packages Scikit-learn 1.5.2, which contains implementations of selected metrics, Keras 2.10, and Tensorflow-gpu 2.10 for deep learning optimization, Numpy 1.26 for matrices and tensors operations, Matplotlib 3.9.2 for plotting. We have also used the R language 4.3.1 for additional data processing of results. For the neural network training, we used the following parameters in the SGD algorithm: a starting learning rate of 10^{-4}, a total of 1200 epochs, and after every 400 epochs, the learning rate was multiplied by 0.1 to decrease it.

Our experiment aimed to verify the effectiveness of the approximation models described in Section 2.3 when trained with data calculated using the method described in Section 2.2. We performed leave-one-out cross-validation by dividing the data set into a training and validation part. We removed individual thicknesses 6, 8, 10, 12, 14, and 16 from the training set and created a test set containing only data related to these excluded thicknesses. We did not make a test set for thicknesses 4 and 18 because we did not want to perform extrapolation. We thus created six pairs of training and validation data sets, for example: the training set contained data related to all thicknesses except 6 mm, extended by probabilistic sampling for each thickness $[4,8,12,14,16,18]$ by $6\cdot 50$ random samples, and the test set contained data related to thicknesses of 6 mm and so on. We trained each of the six neural networks described in Section 2.3, and from these networks, we selected the one that minimizes the MSE value for a given thickness in the training set. The "best" network found in this way is then used to predict values from the test set of residual velocity values. Predictions are made for thicknesses that were not in the training set of this network. The predictive performance is quantified using the MSE, SMAPE, and R2 metrics, considering only data points with values exceeding v_{bl} by at least 10%. This criterion excludes points in proximity to v_{bl}, which, owing to their higher uncertainty, do not adequately capture the characteristics of the anticipated experimental curve.

The network is then used to predict the ballistic limit velocity $\widehat{v_{bl}}$. The values of $\widehat{v_{bl}}$ are compared with the actual values of v_{bl} using the RE and AE metrics. We also test the performance of our proposed probabilistic sampling method depending on different values of the parameters $[c_1,c_2,c_3]$. We tested the same six configurations that we discussed in Section 2.2 and presented in Fig. 4. So, we conducted a total of $6\cdot 6\cdot 6$ experiments, the results of which we aggregated with respect to the parameter values $[c_1,c_2,c_3]$. The results for individual parameters of the probabilistic sampling method are presented in Tables 2, 3, 4, 5, 6, 7.

Table 2

The performance of the best DNN network (see Section 2.3) is according to minimizing MSE on the training dataset. Results were calculated with a leave-one-out cross-validation approach as described in Section 3. The stochastic sampling parameters (4): $[0,0,1]$

Thickness	MSE	SMAPE	R2	v_{bl}	$\widehat{v_{bl}}$	RE	AE
6	1115.017	0.742	0.967	400	413	0.032	13
8	208.943	0.168	0.991	450	452	0.004	2
10	843.319	0.722	0.973	490	496	0.012	6
12	1047.394	0.718	0.958	530	535	0.009	5
14	4384.130	1.209	0.881	570	595	0.044	25
16	11533.472	1.215	0.643	645	619	0.040	26

Table 3

The performance of the best DNN network (see Section 2.3) is according to minimizing MSE on the training dataset. Results were calculated with a leave-one-out cross-validation approach as described in Section 3. The stochastic sampling parameters (4): $[0,1,0]$

Thickness	MSE	SMAPE	R2	v_{bl}	$\widehat{v_{bl}}$	RE	AE
6	575.002	0.107	0.991	400	392	0.020	8
8	1871.402	0.124	0.980	450	434	0.036	16
10	112.547	0.054	0.998	490	482	0.016	8
12	77.787	0.063	0.999	530	516	0.026	14
14	3029.294	0.157	0.932	570	578	0.014	8
16	8392.437	0.400	0.795	645	602	0.067	43

Table 4

The performance of the best DNN network (see Section 2.3) is according to minimizing MSE on the training dataset. Results were calculated with a leave-one-out cross-validation approach as described in Section 3. The stochastic sampling parameters (4): $[0,0.(3),0.(6)]$

Thickness	MSE	SMAPE	R2	v_{bl}	$\widehat{v_{bl}}$	RE	AE
6	895.697	0.294	0.987	400	403	0.007	3
8	2164.763	0.167	0.978	450	445	0.011	5
10	160.639	0.049	0.998	490	492	0.004	2
12	54.851	0.098	0.999	530	530	0.000	0
14	4056.827	0.222	0.914	570	584	0.025	14
16	9407.316	0.520	0.793	645	612	0.051	33

Table 5

The performance of the best DNN network (see Section 2.3) is according to minimizing MSE on the training dataset. Results were calculated with a leave-one-out cross-validation approach as described in Section 3. The stochastic sampling parameters (4): $[1,0,0]$ (uniform distribution)

Thickness	MSE	SMAPE	R2	v_{bl}	$\widehat{v_{bl}}$	RE	AE
6	933.257	0.077	0.983	400	388	0.030	12
8	2176.813	0.082	0.973	450	415	0.078	35
10	117.891	0.023	0.998	490	457	0.067	33
12	118.588	0.034	0.998	530	495	0.066	35
14	2492.610	0.086	0.929	570	545	0.044	25
16	7417.893	0.201	0.703	645	567	0.121	78

We have also prepared detailed graphs for thicknesses of 8 mm and 16 mm, as these are experimental data (not calculated using FEM). In Fig. 6, we present the Recht-Ipson curve plot for 8 and 16 mm, actual (experimental) data for these thicknesses, and ballistic curve predictions made by a network that minimized MSE on the training set for stochastic learning parameters $[0.98,0,0.02]$. In Fig. 3, we present a surface plot of

Table 6

The performance of the best DNN network (see Section 2.3) is according to minimizing MSE on the training dataset. Results were calculated with a leave-one-out cross-validation approach as described in Section 3. The stochastic sampling parameters (4): $[0.(3),0.(3),0.(3)]$

Thickness	MSE	SMAPE	R2	v_{bl}	$\widehat{v_{bl}}$	RE	AE
6	835.399	0.200	0.988	400	405	0.012	5
8	2005.270	0.105	0.978	450	446	0.009	4
10	104.946	0.061	0.998	490	491	0.002	1
12	71.471	0.072	0.999	530	526	0.008	4
14	2757.843	0.166	0.935	570	583	0.023	13
16	8151.024	0.367	0.801	645	609	0.056	36

Table 7

The performance of the best DNN network (see Section 2.3) is according to minimizing MSE on the training dataset. Results were calculated with a leave-one-out cross-validation approach as described in Section 3. The stochastic sampling parameters (4): $[0.98,0,0.02]$

Thickness	MSE	SMAPE	R2	v_{bl}	$\widehat{v_{bl}}$	RE	AE
6	474.367	0.094	0.992	400	410	0.025	10
8	2014.390	0.057	0.975	450	452	0.004	2
10	176.367	0.044	0.996	490	496	0.012	6
12	120.304	0.067	0.998	530	536	0.011	6
14	5157.080	0.187	0.854	570	611	0.072	41
16	7108.285	0.193	0.718	645	616	0.045	29

residual velocity as a function of initial velocity and thickness calculated on the same DNN as in Fig. 6. Predictions were made with a step of 0.25 mm to show the smoothness of the approximated surface.

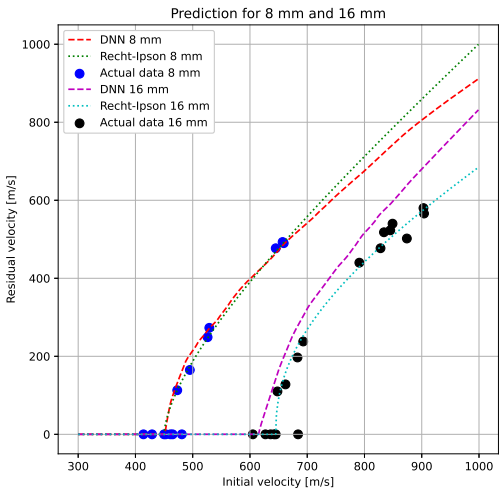


Fig. 6. The Recht-Ipson curve plot for 8 and 16 mm, actual (experimental) data for these thicknesses, and ballistic curve predictions made by a network that minimized MSE on the training set for stochastic learning parameters (4) $[0.98,0,0.02]$

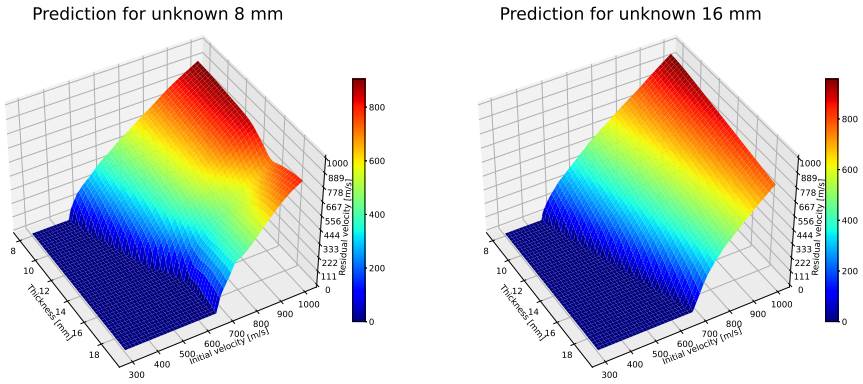
4. DISCUSSION

Based on the results presented in Section 3, we can see that the stochastic sampling method we propose allows for the effective training of a deep neural network algorithm to approximate the ballistic curve for unknown (not present in the training dataset) thickness values. Among the various DNN configurations tested and described in Section 2.3, all obtained similar results. Based on the calculations performed, we cannot determine which one is best for the problem we are investigating. However, our experiments clearly showed the impact of the selection of parameters $[c_1, c_2, c_3]$. In most cases, the values of MSE, SMAPE, and R2 correlated with each other, meaning a decrease in MSE also led to a proportional reduction in SMAPE and R2. Of course, the correlation is not entirely linear because MAE is an absolute error, while SMAPE is a relative error and R2 indicates the proportion of the variance in the dependent variable that is predictable from the independent variable. Taking into account the MSE, SMAPE, and R2 metrics, is seemed that the best fit of the ballistic curve as a whole is obtained when the distribution of stochastic samples is close to uniform sampling (parameters $[1,0,0]$). However, it turned out that adding components related to the first and second derivatives to the CDF mixture can further reduce the fitting error. For 6 mm with parameters [0.98 0 0.02], the MSE was 474.367, the SMAPE was 0.094, and the R2 was 0.992.

For 8 mm $[0,0,1]$ MSE 208.943, SMAPE 0.168, and R2 0.991. For 10 mm with parameters $[0.(3),0.(3),0.(3)]$ MSE was 104.946, SMAPE 0.061, and R2 0.998. For 12 mm with parameters $[0,0.(3),0.(6)]$ MSE was 54.851, SMAPE 0.098, and R2 0.999. For 14 mm with parameters $[1,0,0]$, MSE was 2492.610, SMAPE was 0.086, and R2 was 0.929. For 14 mm with parameters $[0.98,0,0.02]$, MSE was 7108.285, SMAPE was 0.193, and R2 was 0.718. It can therefore be concluded that adding information about the first and second derivatives to the CDF mixture has a decisive impact on the effectiveness of fitting the ballistic curve to the data and the possibility of approximation to unknown thicknesses.

Thanks to the use of the ReLu activation function in the last layer of the tested networks, even though they were not trained on data with values $v_i < v_{bl}$, DNN predictions have values practically equal to zero in the area $v_i < \widehat{v_{bl}}$. This allowed us to apply the heuristic for calculating $\widehat{v_{bl}}$ described in Section 2.3. Using only a uniform distribution significantly worsens the predictive capabilities of ballistic limit velocity. The highest RE and AE (meaning the worst predictive capabilities) were obtained for the $[1,0,0]$ configuration, where only uniform sampling is used. The best prediction results, where $\widehat{v_{bl}}$ is closest to v_{bl}, were obtained for 6 mm for $[0,0.(3),0.(6)]$ where RE = 0.007 and AE = 3, for 8 mm for $[0,0,1]$ and $[0.98,0,0.02]$ where RE = 0.004 and AE = 2, for 10 mm for $[0.(3),0.(3),0.(3)]$ where RE = 0.002 and AE = 1, for 12 mm for $[0,0.(3),0.(6)]$ where RE = 0.000 and AE = 0, for 14 mm for $[0,1,0]$ where RE = 0.014 and AE = 8, for 16 mm for $[0,0,1]$ where RE = 0.040 and AE = 26. The RE and AE errors for all methods using the first and second derivatives in the mixture are relatively low and comparable to a typical experimental error.

The above discussion based on numerical data is very well reflected in Fig. 6. It shows a very accurate estimate of $\widehat{v_{bl}}$ for 8 mm and 16 mm for parameters (4) equal to $[0.98,0,0.02]$, which is RE = 0.004, AE = 2 for 8 mm and RE = 0.045, AE = 29 for 16 mm, respectively. In the case of the approximation of the entire 8 mm curve, the error according to Table 7 is MSE 2014.390, SMAPE 0.057, and R2 0.975, indicating a relatively low level. In the case of 16 mm, the error is MSE 7108.285, SMAPE 0.193, and R2 0.718, which is greater than in the case of 8 mm. This is because the shape of the ballistic curve for 16 mm, with an increase in initial velocity, has different characteristics than the curves generated by FEM for similar thicknesses. The experimental curve for 8 mm has similar characteristics to the FEM data. For this reason, it is obvious that if we trained the DNN without the participation of 16 mm, the quality of the prediction of the entire curve will be worse for cases that deviate more from the training data. This behavior of the approximating model is therefore fully justified. This does not change the fact that the results we obtained are satisfactory. This conclusion is also confirmed in Fig. 7. For both surface plots created for models trained without 8 mm and 16 mm at parameters (4) equal to $[0.98,0,0.02]$, we obtain a smooth surface without disconti-



(a) (b)

Fig. 7. Surface plot of residual velocity as a function of initial velocity and thickness calculated by a network that minimized MSE on the training set for stochastic learning parameters (4) $[0.98,0,0.02]$ trained on data without information about 8 mm thickness (a), and on data without information about 16n mm thickness (b). Predictions were made with a step of 0.25 mm to show the smoothness of the approximated surface

nuities and singularities. However, it is clear that if the dataset contains data for 16 mm (see Fig. 7a), a characteristic depression is visible for 16 mm and its neighbors, resulting from the characteristics of the experimental data. If the approximation model was trained without this data, the depression does not occur because the values are approximated by neighboring values from the training set (see Fig. 7b). This leads to an important conclusion that in ballistics, data obtained from FEM, even if the model was calibrated for a specific experiment, should be treated with caution and cannot replace experimental data entirely. In our experiment, however, this does not significantly affect the demonstration of the satisfactory approximation capabilities of the method we propose, both for predicting the entire ballistic curve and the values of ballistic limit velocities. The evaluation results we obtained (numerical error values) are very similar to those published in earlier works discussed in Section 1.1. However, the use of leave-one-out cross-validation, which other authors did not employ, better motivates the predictive capabilities of our approach.

4.1. Applications and limitations of the proposed method

The stochastic oversampling proposed in this paper is not limited to the Recht–Ipson equation; we anticipate that it can also be used to generate samples for any phenomenon with an analytical (physical-based) or engineering (experimental-based) equation describing the observed phenomenon. The method proposed in this work should be used in particular cases, when we want to build a numerical (machine learning-based) extension of our method to additional dimensions that do not have counterparts in the analytical formula. In such an extension, the approximation may uncover new physical understanding. Based on known laws of physics and engineering formulas, we can guarantee that the newly generated data meets the assumptions supported by these laws and experience for the examined ranges of input arguments. After developing new samples using stochastic-based oversampling, we can use them to train a method to approximate the phenomenon under study, accounting for parameters not used in the original set, such as material thickness in the case of ballistic events. The numerical method we can use for approximation is not limited to neural networks – in practice, we can use any method that allows nonlinear approximation, such as a Gaussian process, a random forest, or a support vector machine. Our choice of a multilayer feedforward neural network as an approximator was motivated by universal approximation theorems [36].

In our use case, i.e., the approximation of two-dimensional ballistic limit curves, accurate estimation of the slope characteristic near v_{bl} is subject to uncertainty, as shown in Fig. 3a. For this reason, training an effective method for approximating the terminal ballistic curve over its entire length above v_{bl} is practically impossible with the small number of samples that occur in real ballistic experiments. Therefore, the use of some form of oversampling, such as uniform sampling or an advanced method like ours, is necessary to do so.

The idea of our approach is similar to that of physics-informed neural networks. PINNs incorporate the laws of physics relevant to a given phenomenon into the loss function to reduce the search space during optimization. However, like any other approximation method, such networks cannot effectively fit the data without a sufficiently large training set. Our method is therefore a natural complement to PINNs: it allows the training set to be expanded with additional points that not only meet the physical assumptions of a given phenomenon but are also distributed in such a way that it is possible to effectively map the studied function in the area where it is subject to the greatest variability. Therefore, the stochastic physical-based oversampling proposed in this work is not a standalone method—it requires an additional approximation method (usually nonlinear) that can cover the solution space.

4.2. Determining the impact of oversampling parameter selection on approximator training results

An interesting question is whether and to what extent selecting different oversampling parameters affects the results of approximation training. In other words, are there statistically significant differences in the performance of networks trained for different parameters $[c_1, c_2, c_3]$ in (3). To perform this test, we conducted a statistical analysis by examining the confidence interval for the difference in means between predictions for methods trained on data without information about 8 mm thickness and validated on 8 mm thickness across all parameter configurations (3) (see Fig. 4). We used the absolute differences between the predicted and actual values on BLC as the random variables. Since the distribution of this variable is not normal, we used the pairwise Bootstrap, a nonparametric statistical technique. We set the confidence coefficient at 0.95 and the Bootstrap sample size at 10^5. The results of the pairwise comparisons are presented in Fig. 8. Since the confidence interval for the means of two random variables is symmetric, the plots are presented as a triangle over the diagonal. We interpret those plots as follows: if 0 does not belong to the confidence interval, we conclude that there is a statistically significant difference between two distributions, which means that the change in the oversampling configuration of the training dataset resulted in significantly different error values generated during prediction on the test set. If 0 belongs to the confidence interval, we conclude that there is no statistically significant difference between two distributions. According to the calculations, there is a statistically significant difference between the mean absolute values of the difference between predicted and actual residual velocities with oversampling in all tested configurations except $[0.0(3), 0.0(6)]$ and $[0.98, 0.0, 0.2]$; $[0.1, 0]$ and $[0.(3), 0.(3), 0.(3)]$; $[0.1, 0]$ and $[0.98, 0, 0.2]$; $[0.(3), 0.(3), 0.(3)]$ and $[0.98, 0, 0.2]$. It is worth noting that a statistically significant difference always occurs for the extreme cases, i.e., $[1, 0, 0]$ and $[0, 0, 1]$, in which we have uniform sampling or sampling based on the second derivative of the function, respectively. The other cases have very similar CDFs, so it is not surprising that some pairs draw similar sample distributions to the training set, meaning that the networks are trained on almost the same data set and perform similarly. These results confirm the impact of oversampling parameters on the training results of the approximator.

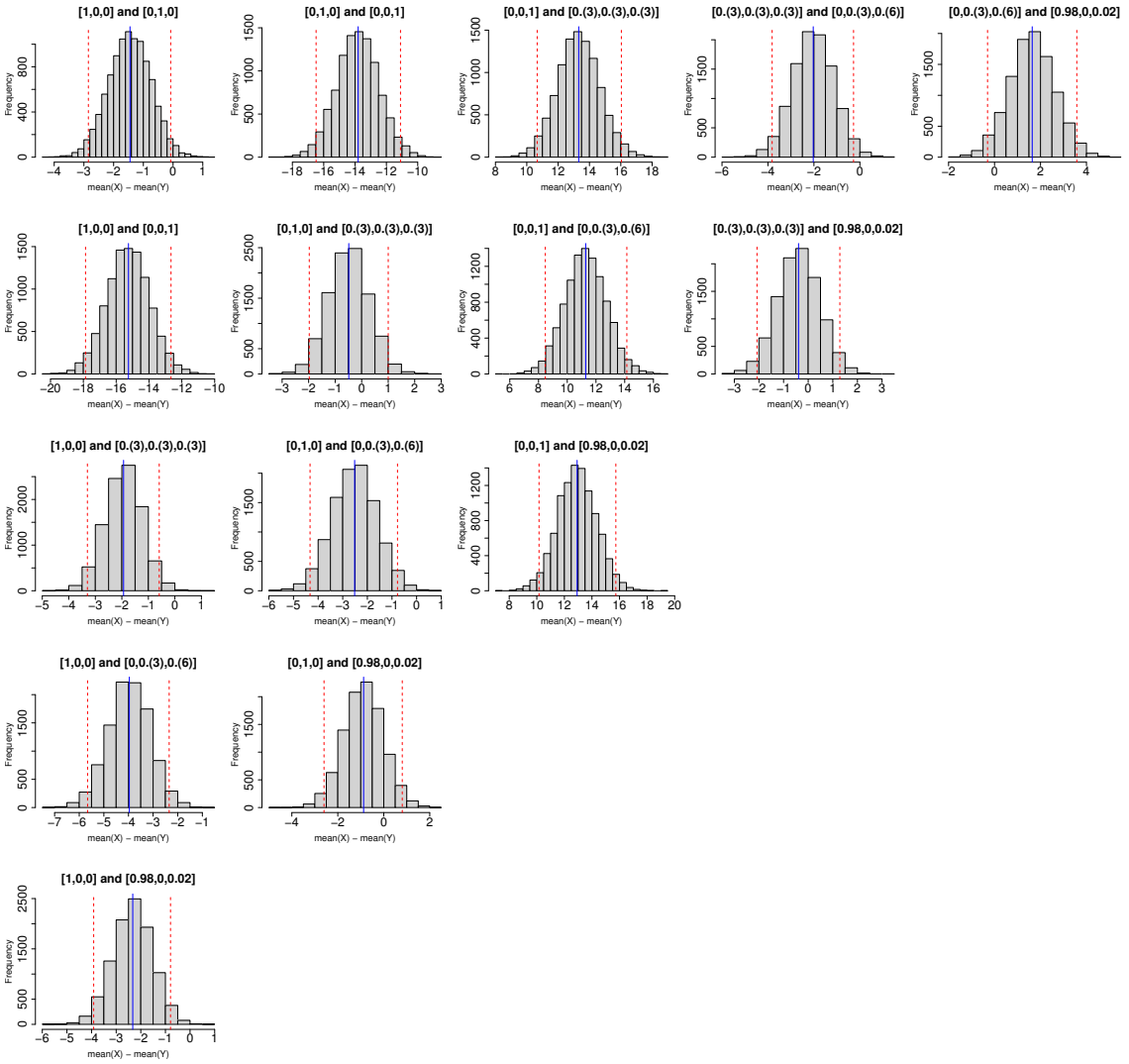


Fig. 8. Histogram visualization of Bootstrap statistical analysis examining the confidence interval for the difference in means between predictions for methods trained on data without information about 8 mm thickness and validated on 8 mm thickness across all parameter configurations (3) (see Fig. 4). We used the absolute differences between the predicted and actual values on BLC as the random variables. We set the confidence coefficient at 0.95 and the Bootstrap sample size at 10^5. Since the confidence interval for the means of two random variables is symmetric, the plots are presented as a triangle over the diagonal. Blue lines indicate differences of means, while red lines are confidence intervals with 0.95 confidence

5. CONCLUSION

As shown by the results presented in Section 3 and discussed in Section 4, the stochastic physical-based oversampling method presented in this paper allows for effective approximation of two-dimensional ballistic limit curves in the range of material thickness. The FEM-based experiments were calibrated to ensure their consistency with measurements obtained from experiments. The use of FEM to calculate results for certain thicknesses was a trade-off between the accuracy of the laboratory experiment and the significant cost of conducting it. To test the consistency of our prediction with the experimental results, we performed leave-one-out validation, in which we estimated the values of ballistic limit curves for 8 mm and 16 mm (see Fig. 7), obtained from laboratory experiments. In this way, we showed that our results agree with real-world results. Of course, the values plotted in Fig. 7 for thicknesses between those in our dataset, i.e., 4 mm, 6 mm, 8 mm, 12 mm, 14 mm, 16 mm, and 18 mm (e.g., 7 mm) are approximations. We cannot assess their accuracy, nor do we do so in our study. However, for values in the range 6 mm, 8 mm, 10 mm, 12 mm, 14 mm, and 16 mm we can reliably estimate the approximation's accuracy, as we did and described in Sections 3 and 4. The study aimed to present the data-treatment method and to demonstrate the applicability and effectiveness of the machine-learning approach used. In an idealized world, all data would be obtained from experiments that are free of experimental-setup limitations, operator influence, sample-geometry and heterogeneity, boundary and environmental conditions, and so on. For now, however, the quality of the output is as good as the quality of the input data. What the obtained results demonstrated is that the chosen algorithmic im-

plementation is capable of treating the given dataset efficiently and would therefore be able to work on any other similar or enhanced data. Our experiments have shown that it is possible not only to approximate the shape of the curve but also to accurately predict the ballistic limit velocity for material thicknesses not present in the dataset. The inclusion of information about the first and second derivatives in the stochastic oversampling process allowed for a significant increase in prediction accuracy over uniform sampling. Oversampled datasets prepared in this way allow for effective training of machine learning algorithms such as deep neural networks. As we have shown, our method can use a small number of samples from experimental data and FEM simulations to create accurate two-dimensional approximations of the relationship between residual velocity, input velocity, and thickness. In practice, this allows for a significant reduction in the cost of conducting a complete ballistic evaluation of the material for different thicknesses. This is a very important practical application of our method. Our method belongs to a group of algorithms that use known physical properties described by analytical formulas. These formulas can be used to create a machine learning model that generalizes the description of the analyzed phenomenon to subsequent dimensions, as in the case of ballistic curves with variable thickness. We do not consider physical laws in the function minimized by the algorithm, as in the physical-informed neural networks (PINNs) approach, but instead integrate them directly into the data. This eliminates the need to adapt the machine learning training algorithm and loss function to a specific problem.

We also believe that our approach is not limited to ballistic curves but extends to other one- or multi-dimensional problems described by an analytical formula, which can be expanded by additional dimensions not included in the formula. However, such applications require further research, which we plan to carry out in the future.

REFERENCES

[1] L. Lomazzi, D. Morin, F. Cadini, A. Manes, and V. Aune, "Deep learning-based analysis to identify fluid-structure interaction effects during the response of blast-loaded plates," *Int. J. Prot. Struct.*, vol. 15, pp. 722–752, 2023, doi: 10.1177/20414196231198259.

[2] A. Sung, "Ranking importance of input parameters of neural networks," *Expert Syst. Appl.*, vol. 15, no. 3, pp. 405–411, 1998, doi: 10.1016/S0957-4174(98)00041-4.

[3] S. Ryan *et al.*, "Machine learning for predicting the outcome of terminal ballistics events," *Def. Technol.*, vol. 31, pp. 14–26, 2024, doi: 10.1016/j.dt.2023.07.010.

[4] B. Tahenti, F. Coghe, and R. Nasri, "Ballistic limit estimation approaches for ballistic resistance assessment," *Def. Sci. J.*, vol. 70, no. 1, pp. 82–89, 2020.

[5] S. Thompson, F. Teixeira-Dias, M. Paulino, and A. Hamilton, "Ballistic response of armour plates using generative adversarial networks," *Def. Technol.*, vol. 18, no. 9, pp. 1513–1522, 2022, doi: 10.1016/j.dt.2021.08.001.

[6] Y. Deng, X. Yang, and X. Huang, "Determination of ballistic resistance model of finite thickness material based on artificial neural network," *Thin-Walled Struct.*, vol. 203, p. 112161, 2024, doi: 10.1016/j.tws.2024.112161.

[7] S. Cuomo, V.S. Di Cola, F. Giampaolo, G. Rozza, M. Raissi, and F. Piccialli, "Scientific machine learning through physics–informed neural networks: Where we are and what's next," *J. Sci. Comput.*, vol. 92, no. 3, p. 88, 2022, doi: 10.1007/s10915-022-01939-z.

[8] Z.K. Lawal, H. Yassin, D.T.C. Lai, and A. Che Idris, "Physics-informed neural network (pinn) evolution and beyond: A systematic literature review and bibliometric analysis," *Big Data Cogn. Comput.*, vol. 6, no. 4, p. 140, 2022, doi: 10.3390/bdcc6040140.

[9] S. Ryan, H. Le, J. Berk, A.A. Kumar, and S. Venkatesh, "Physics-informed machine learning for predicting the ballistic limit of whipple shields," *Int. J. Impact Eng.*, vol. 203, p. 105364, 2025, doi: 10.1016/j.ijimpeng.2025.105364.

[10] T. Hachaj, Ł. Bibrzycki, and M. Piekarczyk, "Fast training data generation for machine learning analysis of cosmic ray showers," *IEEE Access*, vol. 11, pp. 7410–7419, 2023, doi: 10.1109/ACCESS.2023.3237800.

[11] J. Fan and I. Gijbels, "Data-driven bandwidth selection in local polynomial fitting: variable bandwidth and spatial adaptation," *J. R. Stat. Soc. B*, vol. 57, no. 2, pp. 371–394, 1995, doi: 10.1111/j.2517-6161.1995.tb02034.x.

[12] A. Sudou, P. Hartono, R. Saegusa, and S. Hashimoto, "Signal reconstruction from sampled data using neural network," in *Proc. 12th IEEE Workshop on Neural Networks for Signal Processing*, 2002, pp. 707 – 715, doi: 10.1109/NNSP.2002.1030082.

[13] B. Adcock and N. Dexter, "The gap between theory and practice in function approximation with deep neural networks," *SIAM J. Math. Data Sci.*, vol. 3, no. 2, pp. 624–655, 2021.

[14] P. Grohs and F. Voigtlaender, "Proof of the theory-to-practice gap in deep learning via sampling complexity bounds for neural network approximation spaces," *Found. Comput. Math.*, vol. 24, no. 4, pp. 1085–1143, 2024.

[15] B. Adcock, S. Brugiapaglia, N. Dexter, and S. Moraga, "Chapter 1 – learning smooth functions in high dimensions: From sparse polynomials to deep neural networks," in *Numerical Analysis Meets Machine Learning*, ser. Handbook of Numerical Analysis, S. Mishra and A. Townsend, Eds. Elsevier, 2024, vol. 25, pp. 1–52, doi: 10.1016/bs.hna.2024.05.001.

[16] B. Adcock, J.M. Cardenas, and N. Dexter, "An adaptive sampling and domain learning strategy for multivariate function approximation on unknown domains," *SIAM J. Sci. Comput.*, vol. 45, no. 1, pp. A200–A225, 2023, doi: 10.1137/22M1472693.

[17] E. Flores-Johnson, M. Saleh, and L. Edwards, "Ballistic performance of multi-layered metallic plates impacted by a 7.62-mm APM2 projectile," *Int. J. Impact Eng.*, vol. 38, pp. 1022–1032, 2011, doi: 10.1016/j.ijimpeng.2011.08.005.

[18] J.J. Moré, B.S. Garbow, and K.E. Hillstrom, "User guide for MINPACK-1," CM-P00068642, Tech. Rep., 1980, doi: 10.2172/6997568.

[19] L.S. Lasdon, R.L. Fox, and M.W. Ratner, "Nonlinear optimization using the generalized reduced gradient method," *Rev. Fr. Autom. Inform. Rech. Oper.*, vol. 8, no. V3, pp. 73–103, 1974, doi: 10.1051/ro/197408V300731.

[20] R.F. Recht and T.W. Ipson, "Ballistic perforation dynamics," *J. Appl. Mech.*, vol. 30, no. 3, pp. 384–390, 1963, doi: 10.1115/1.3636566.

[21] "Impetus AFEA - User's manual," 2025, accessed: 2025-06-25. [Online]. Available: https://www.impetus.no/support/manual/

[22] T. Fras, A. Bracq, M. Seidl, and G. Vincent, "Experimental and numerical investigation on the ricochet of hard-core 7.62 small-calibre projectiles from the SiC and Al2O3 ceramic targets," *Int. J. Impact Eng.*, vol. 192, p. 105010, 2024, doi: 10.1016/j.ijimpeng.2024.105010.

[23] T. Fras, "On the effect of pitch and yaw angles in oblique impacts of small-caliber projectiles," *Def. Technol.*, vol. 31, pp. 73–94, 2024, doi: 10.1016/j.dt.2023.06.004.

[24] G.R. Johnson, "A constitutive model and data for metals subjected to large strains, high strain rates and high temperatures," in *Proc. 7th International Symposium on Ballistics, The Hague, Netherlands, 1983*, 1983, pp. 541–547.

[25] G.R. Johnson and W.H. Cook, "Fracture characteristics of three metals subjected to various strains, strain rates, temperatures and pressures," *Eng. Fract. Mech.*, vol. 21, no. 1, pp. 31–48, 1985.

[26] T. Fras, I. Szachogluchowicz, and L. Sniezek, "Ti6al4v-aa1050-aa2519 explosively-cladded plates under impact loading," *Eur. Phys. J. Spec. Top.*, vol. 227, no. 1, pp. 17–27, 2018.

[27] T. Fras, C.C. Roth, and D. Mohr, "Application of two fracture models in impact simulations," *Bull. Pol. Acad. Sci., Tech. Sci.*, vol. 68, no. 2, 2020.

[28] M. Stanczak, A. Rusinek, P. Broniszewska, T. Fras, and P. Pawłowski, "Influence of strain rate and temperature on the mechanical behaviour of additively manufactured alsi10mg alloy–experiment and the phenomenological constitutive modelling," *Bull. Pol. Acad. Sci., Tech. Sci.*, vol. 70, no. 4, 2022.

[29] P. Simon, "Modélisation du comportement mécanique et de la rupture en conditions dynamiques d'aciers de structure et à blindage. Manuscrit de thèse," 2019. [Online]. Available: https://hal.univ-lorraine.fr/tel-02327889

[30] J. Ribeiro, A. Santiago, and C. Rigueiro, "Damage model calibration and application for S355 steel," *Procedia Struct. Integr.*, vol. 2, pp. 656–663, 2016.

[31] X. An, A. Artemyev, V. Angelopoulos, S. Lu, P. Pritchett, and V. Decyk, "Fast inverse transform sampling of non-gaussian distribution functions in space plasmas," *J. Geophys. Res.-Space Phys.*, vol. 127, no. 5, p. e2021JA030031, 2022.

[32] T. Hachaj, M. Piekarczyk, Ł. Bibrzycki, and J. Wąs, "Determination of spherical coordinates of sampled cosmic ray flux distribution using principal components analysis and deep encoder-decoder network," *Mach. Graph. Vis.*, vol. 33, no. 2, pp. 29–45, 2024, doi: 10.22630/MGV.2024.33.2.2.

[33] R.D. Riley *et al.*, "Minimum sample size for developing a multivariable prediction model: Part ii-binary and time-to-event outcomes," *Stat. Med.*, vol. 38, no. 7, pp. 1276–1296, 2019.

[34] D.G. Jenkins and P.F. Quintana-Ascencio, "A solution to minimum sample size for regressions," *PLoS One*, vol. 15, no. 2, p. e0229345, 2020.

[35] Y. Tian, Y. Zhang, and H. Zhang, "Recent advances in stochastic gradient descent in deep learning," *Mathematics*, vol. 11, no. 3, p. 682, 2023, doi: 10.3390/math11030682.

[36] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Netw.*, vol. 2, no. 5, pp. 359–366, 1989, doi: 10.1016/0893-6080(89)90020-8.