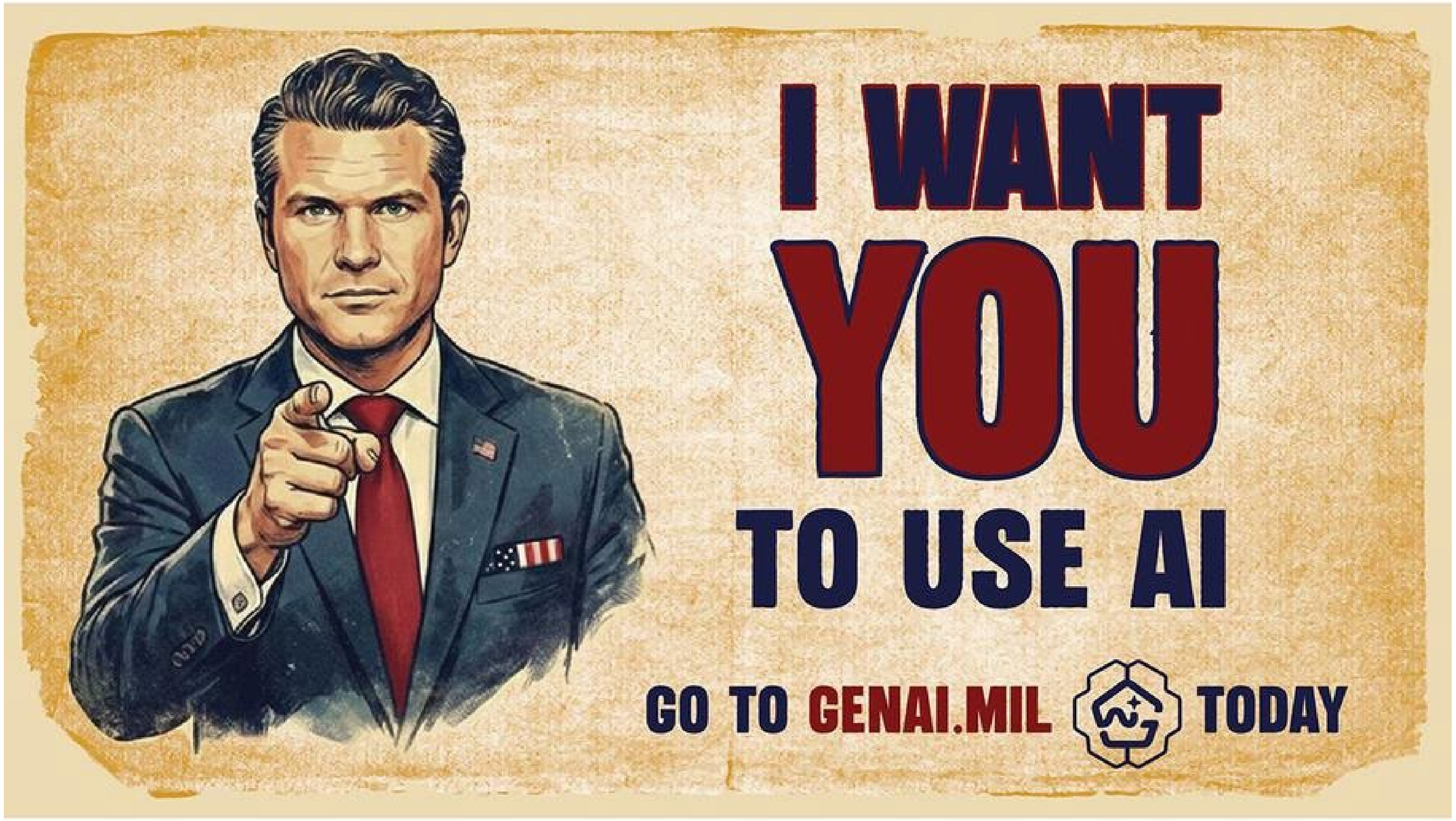
Deep-Learning Based Improvements to Explosives Safety Board Tools

Serdar Astarlioglu, Ph.D.
Safety Engineer
Explosives Safety Office

SLIDES ONLY

NO SCRIPT PROVIDED

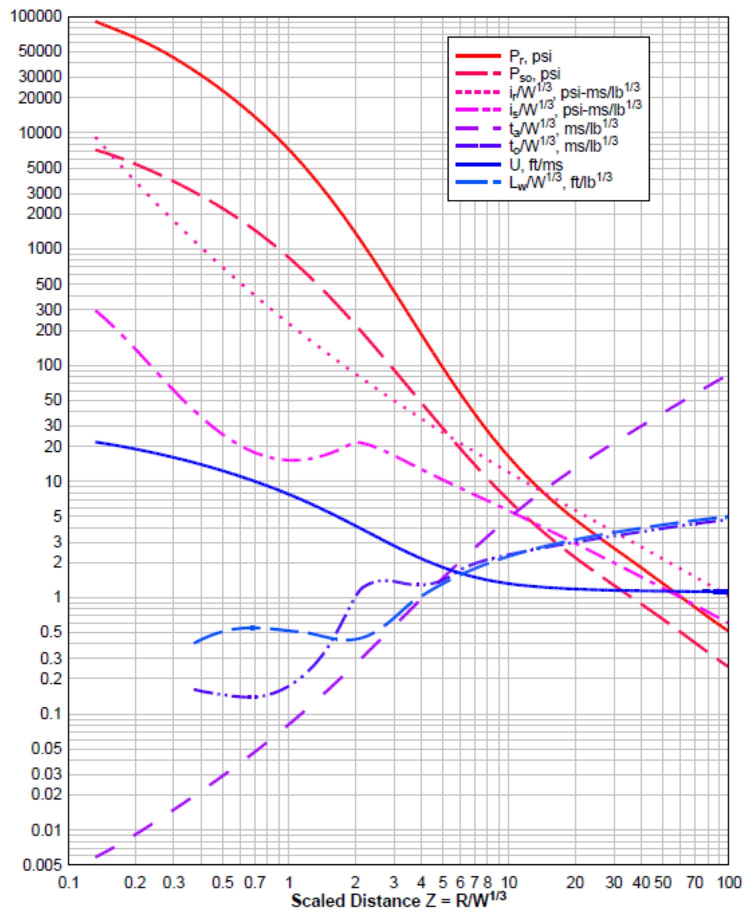
IESSE 25
Phoenix, AZ
January 21-23, 2026


I WANT
YOU
TO USE AI
GO TO GENAI.MIL TODAY

Motivation

- Traditionally, we have been using curve fits to predict consequences of weapon effects
  - Kingery-Bulmash Equations (ConWep)
  - Spall and breach equations in UFC 3-340-01
  - Brode function
  - TP-16 Fits for primary fragments
  - TP-17 Fits for airblast
- These equations typically use one input variable (e.g., NEW).

Example: Kingery-Bulmash Equations

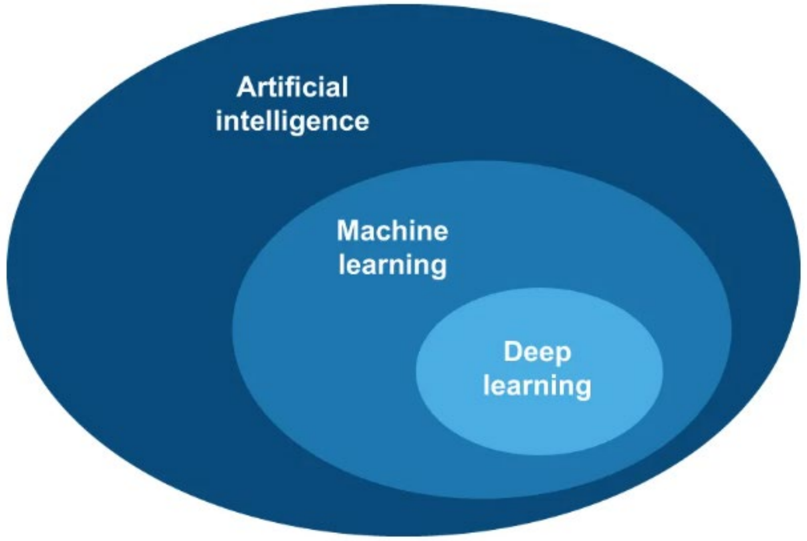


UFC 3-340-02
5 December 2008
Change 2, 1 September 2014

UFC 3-340-02

- Positive phase shock wave parameters for a spherical TNT explosion in free air
- Input: scaled distance
- Output: peak reflected and incident pressures, impulses, shock arrival time, positive phase duration, etc.

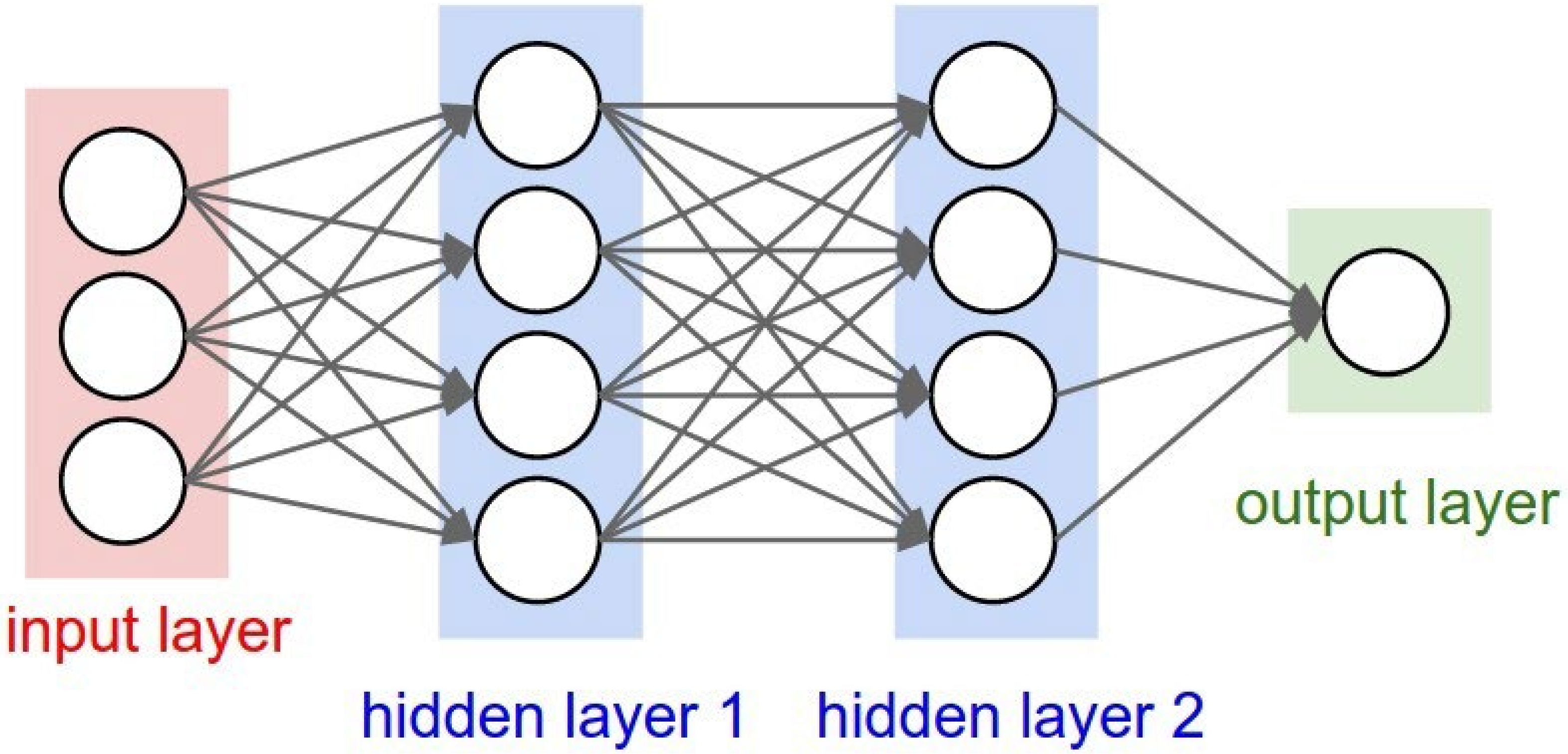
Deep Learning



Ref: Deep Learning with Python
https://deeplearningwithpython.io/

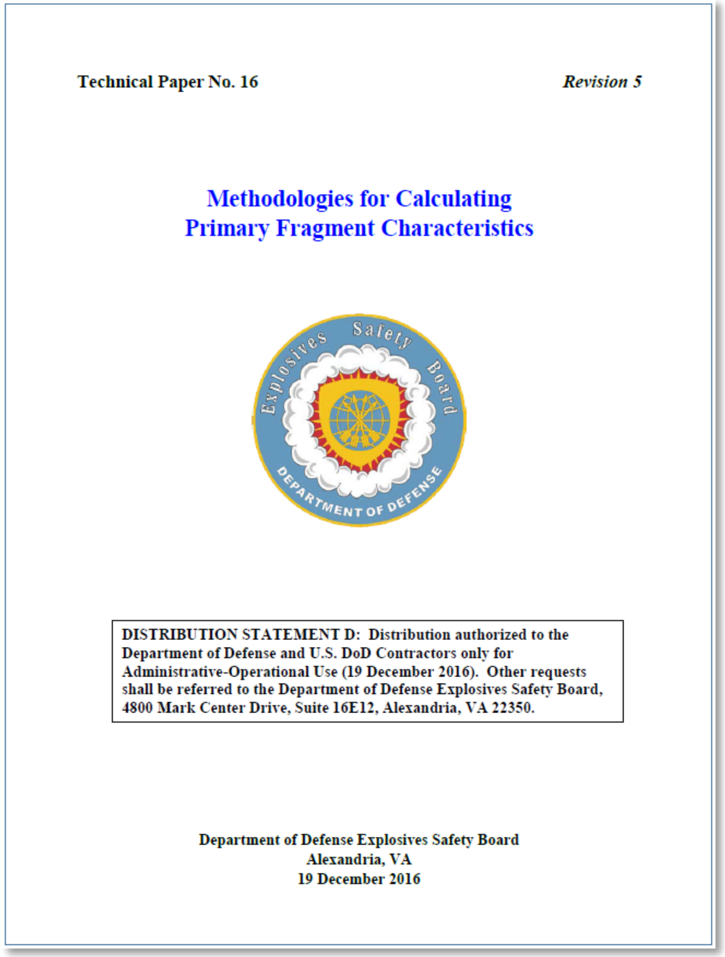
- **Artificial Intelligence (AI)**: Effort to automate intellectual tasks normally performed by humans
- **Machine Learning (ML)**: A new programming paradigm that is trained rather than programmed with explicit rules
- **Deep Learning**: Subset of ML that emphasizes learning successive layers of increasingly meaningful representations (outputs). E.g., given charge weight, case diameter, and case weight, predict the hazardous fragment distance with 95% confidence level

A regular 3-layer Neural Network.



Ref: https://cs231n.github.io/convolutional-networks/#conv

TP-16



Technical Paper No. 16 *Revision 5*

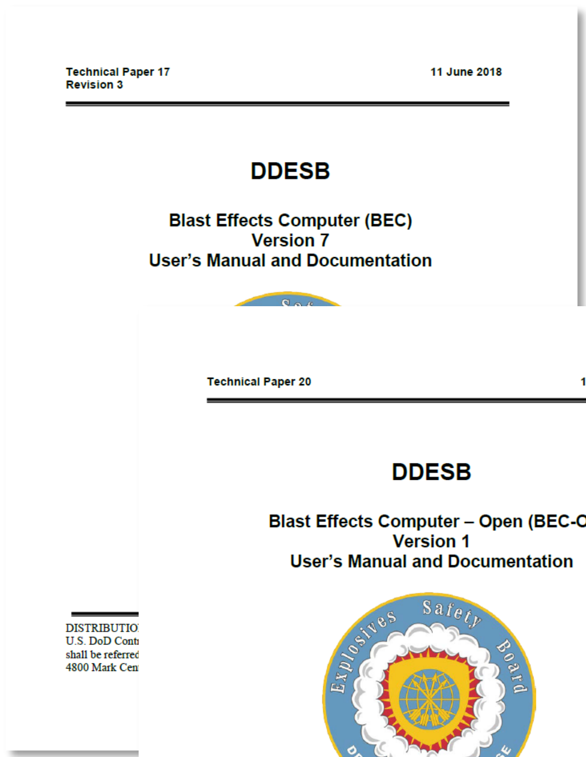
Methodologies for Calculating
Primary Fragment Characteristics

DISTRIBUTION STATEMENT D: Distribution authorized to the Department of Defense and U.S. DoD Contractors only for Administrative-Operational Use (19 December 2016). Other requests shall be referred to the Department of Defense Explosives Safety Board, 4800 Mark Center Drive, Suite 16E12, Alexandria, VA 22350.

Department of Defense Explosives Safety Board
Alexandria, VA
19 December 2016

- TP-16 provides DoW Explosives Safety Board (ESB) approved methodologies for calculating characteristics of primary fragments (primary fragment mass and velocity, hazardous fragment distance, etc.

TP-17



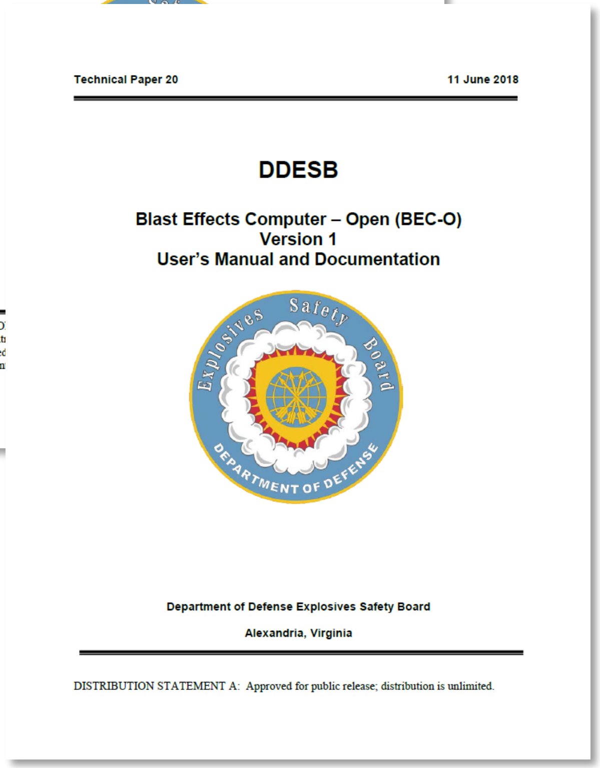
Technical Paper 17
Revision 3

11 June 2018

DDESB

Blast Effects Computer (BEC)
Version 7
User's Manual and Documentation

DISTRIBUTIO
U.S. DoD Cont
shall be referred
4800 Mark Cen



Technical Paper 20

11 June 2018

DDESB

Blast Effects Computer – Open (BEC-O)
Version 1
User's Manual and Documentation

Department of Defense Explosives Safety Board

Alexandria, Virginia

DISTRIBUTION STATEMENT A: Approved for public release; distribution is unlimited.

- TP-17 and TP-20 provide the user manual for Blast Effects Computer (BEC) and Blast Effects Computer – Open (BEC-O)
- Given PES, munition type, NEW, and distance to ES, determine the airblast parameters

Deep Learning Example - Robust AE

Objective: Use AI to predict MFD-H (ft) for a given NEW (lb), diameter (in), and cylindrical case weight (lb). The AI is trained using the same dataset used for fitting TP-16 Eq. 4-1a.

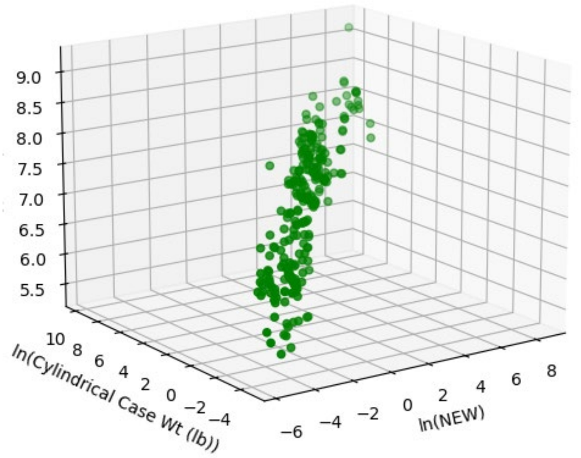
Python Libraries

- *pandas* is used for reading data frame from Excel
- *numpy* is used for arrays and statistical functions
- *PyTorch* is used for machine learning
- *sklearn* is used for splitting dataset into training and testing datasets

The Data Format

- Read the necessary columns from the Excel spreadsheet.
- The approach used here is to convert all the experimental data to log space and train AI using the logs of the actual variables.

I/O	Index	Description
input	1	Single item NEW (lbs)
input	2	Diameter (in)
input	3	Cylindrical Case Weight (lbs)
output	1	MFD-H (ft)



Data Preparation

- The experimental dataset are placed in X (input items are logs of NEW, diameter, case wt) and y (output items are logs of MFD-H) numpy arrays.
- These arrays are then split into train (%80 of data) and test (20% of data) sets.
- The purpose of this split is to check if the AI is being over-trained to memorize the data.
- Ideally, the loss functions for test and train should level off to the same horizontal asymptote. If they are too different, it means the model is memorizing the training set and gives garbage for anything not used in training (i.e., test set).

Define Neural Network Model

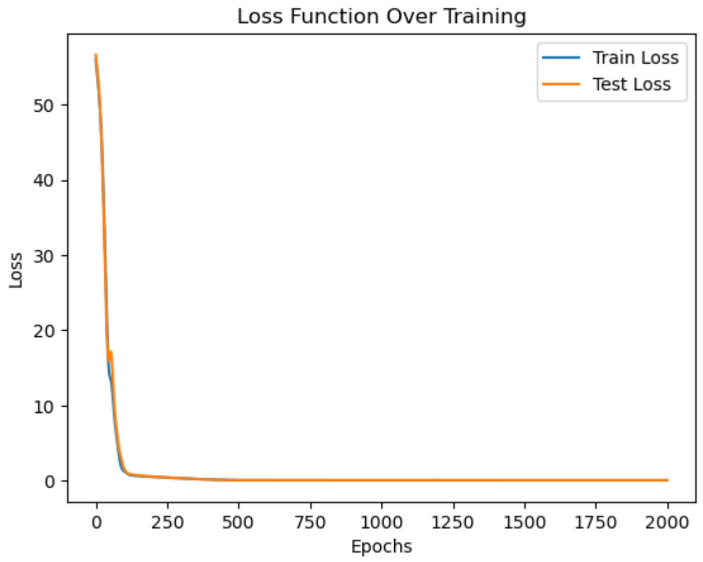
- The model used here has two hidden layers. The number of neurons in the hidden layers can be changed. If there are no hidden layers, the result will be constant.

```
# Define the model
class NeuralNet(nn.Module):
    def __init__(self, n_input = 3, n_output = 1, n_hidden = 10):
        super(NeuralNet, self).__init__()
        self.fc1 = nn.Linear(n_input, n_hidden)     # first hidden layer (n_input)
        self.fc2 = nn.Linear(n_hidden, n_hidden)    # second hidden layer
        self.fc3 = nn.Linear(n_hidden, n_output)    # output layer

    def forward(self, x):
        x = torch.relu(self.fc1(x))
        x = torch.relu(self.fc2(x))
        x = self.fc3(x)
        return x
```

Train the Model

- Create an ML model (i.e. NeuralNet)
- Define criterion (i.e., Mean Squared Error (MSELoss) as loss function)
- Define the optimizer (i.e., Adaptive Moment Estimation (Adam) as iterative optimization algorithm to minimize the loss function)

```
epochs = 2000
train_losses, test_losses = [], []

for epoch in range(epochs):
    # Train
    model.train()
    optimizer.zero_grad()
    y_pred_train = model(X_train_tensor)
    loss_train = criterion(y_pred_train, y_train_tensor)
    loss_train.backward()
    optimizer.step()
```



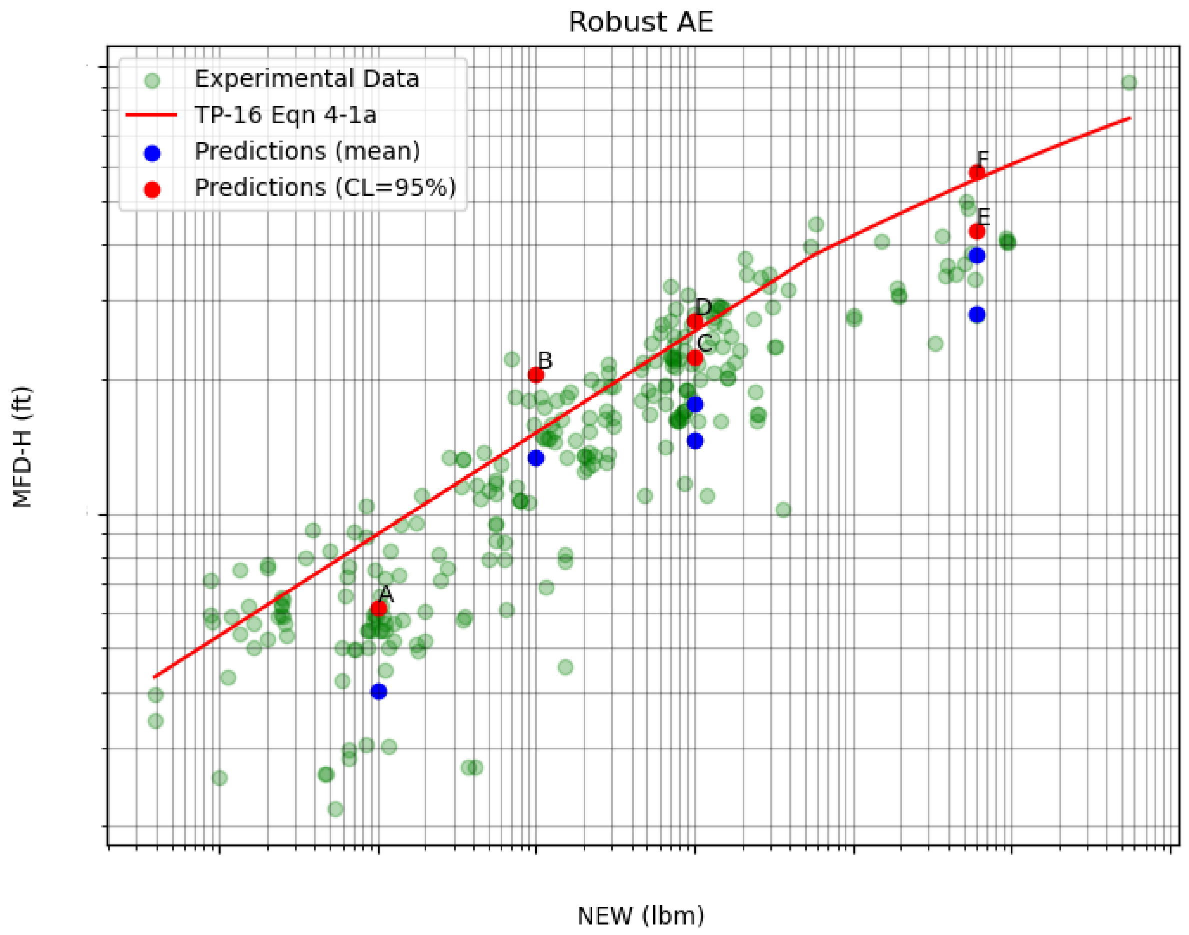
Use Model to Make Predictions

- The input is: NEW (lb), Diameter (in), and Cyl. Case Weight (lb).
- The output is: MFD-H (ft)
- Note that the model was trained on logs of the input values and gives the log of the MFD-H.

Label	NEW (lbs)	Diameter (in)	Cyl. Case Wt. (lbs)
A	0.1	1	0.1
B	1	2	4
C	10	3	10
D	10	1	20
E	600	20	150
F	600	20	350

Note: These values are arbitrary – they are not representative of real A&E

Comparison with TP-16



- TP-16 fit is conservative on average
- The level of conservatism is potentially higher when realistic A&E data is used

Summary

- Simple precursory analysis to determine potential benefits of using AI in ESB tools
- Data is the key
- Multiple input parameters can easily be accommodated(e.g., dividing ammunition into robust, extremely heavy cased, and non-robust categories may not be needed)
- Retrain with ease as new data becomes available
- Not compatible with Excel. AI-based tools need to be provided on a different platform (web-based?)
- Potential changes to TP-16, TP-17, and TP-20