Comparing the Prediction Capabilities of an Artificial Neural Network vs a Phenomenological Model for Predicting the Terminal Ballistics of Kinetic Energy Projectiles

John R. Auten Sr.
jauten@towson.edu

Robert J. Hammell II
rhammell@towson.edu

Department of Computer and Information Sciences
Towson University
Towson, MD 21252, USA

Abstract

The need to accurately predict the terminal ballistics of kinetic energy projectiles in Vulnerability/Lethality models is of paramount importance to the Department of Defense. An artificial neural network was trained on a set of 1,625 data points to predict perforation, residual velocity, and residual mass of a kinetic energy projectile impacting a homogeneous-monolithic-metallic target. The capability of that neural network is analyzed against a phenomenological model that is currently used by the Department of Defense for modeling the terminal ballistics of kinetic energy projectiles. Results show that the neural network model is superior in speed and accuracy to the current phenomenological model.

Keywords: Artificial Neural Network, Kinetic Energy Projectiles, Terminal Ballistics.

1. INTRODUCTION

This paper presents research efforts related to the selection, development, and performance of an Artificial Neural Network (ANN) to predict the terminal ballistics of Kinetic Energy Projectiles (KEPs). This work extends research previously published by Auten & Hammell (2014a, 2014b).

When the Department of Defense (DoD) acquires weapon systems for use in the battlefield by U.S. soldiers, the systems need to be safe and effective. In order to ensure that they are safe and effective, the DoD tests the system and uses modeling and simulation to augment the results from the tests. The DoD requires that Acquisition Category (ACAT) I systems undergo Live-Fire Test & Evaluation (LFT&E) (U.S. Department of Defense, 2008) to determine the Vulnerability/Lethality (V/L) of that system. Simulation models are validated to those live-fire tests and then accredited so that they can be used for future studies involving that system. An important part of V/L simulation models is the terminal ballistics models that are used to determine if a threat has perforated a particular target and what the residual capability of that threat is after perforation.

The purpose of this paper is to present preliminary results from the training of an Artificial Neural Network (ANN) for the prediction of perforation of a monolithic metallic target plate. These results are the first part of a two-step model that is being developed to predict the terminal ballistics of Kinetic Energy Projectiles (KEPs). After prediction perforation has been achieved, the next step will be to develop an ANN that can predict the residual capabilities of a KEP after perforation of a target plate.

Two important factors in the current work include issues with finding data to train the ANN with and choosing a model to which the ANN's performance can be compared. Those two topics will now be briefly discussed in this introduction.

Data Issues

During previous research towards constructing better models, it became very apparent that more data was needed to ensure a better fit for the neural network (Auten & Hammell, 2014a). The task of collecting more data has been hampered by missing data points, a lack of publically available data, poor data recording practices, and a lack of funding to generate new test data.

Recently a database of over 7000 test data records was acquired. The database will require a large effort to make all of the data usable, but it was possible to clean over 1500 of the records and add them to the database for this research. The total count of records available went from 530 to 2125. The additional data provided a better spread over the problem space and for the case of perforation outcome it also provided a better distribution of outcomes.

In the early database of 530 data points the percent of outcomes that were perforation was 82%. With the current database the percent of outcomes that were perforation is lowered to 57%. This all resulted in a more robust training set to use for training the ANN.

Segletes Hybrid Rod Model

A study was performed by the U.S. Army Research Laboratory (ARL) to evaluate the performance of algorithms that are used to model the terminal ballistics of kinetic energy projectiles in Vulnerability/Lethality models. The Segletes hybrid rod model (Segletes, 2000) was selected as performing the best across all of the possible scenarios that would likely be encountered in a Vulnerability/Lethality (V/L) model (Auten, 2012).

Given the performance of the Segletes hybrid rod model in the U.S. ARL study, it was chosen as the model to compare against the artificial neural network that was developed as part of this research effort.

The remainder of the paper is organized as follows: Section 2 describes the process used to develop the ANN for this research. Section 3 describes an analysis of the simulation results of both models run on a test data set. Section 4 presents conclusions, followed by a discussion related to future research.

2. MODEL DEVELOPMENT PROCESS

This section will provide an overview of the process used to develop the ANN model. The first section will provide an overview of the data used by the model, the second section will outline the process used to select the topology of the ANN, and the final section will outline the process used to train the chosen topology.

Data Metrics and Preparation

The neural network has 11 inputs that consist of striking velocity, total yaw, rod density, rod length, rod diameter, rod hardness, target density, target hardness, target thickness, target obliquity, and target Young's modulus. The outputs consist of perforation (-0.9 = non-perforation and 0.9 = perforation), residual velocity, and residual mass. Information regarding the inputs and outputs can be found in Table 1.

	Mean	Std Dev	Minimum	Q1	Median	Q3	Maximum
Striking Velocity (m/s)	1263.5	190.1	580	1139	1275	1390	1841
Total Yaw (°)	0.9	1.1	0	0.4	0.6	1	12.3
Rod Density (g/m^3)	17.5	2.3	7.9	17.3	17.8	18.6	18.6
Rod Length (mm)	109.1	30.7	41.3	95.1	102.2	140	214.2
Rod Diameter (mm)	7.3	2	3.9	5.8	6.8	7.9	15
Rod Hardness (BHN)	407.7	66.7	233.3	374.6	394.8	427.2	869.1
Target Density (g/m^3)	7.8	0.2	2.7	7.9	7.9	7.9	7.9
Target Hardness (BHN)	322.8	79.5	29	269	302	340	555
Taret Thickness (mm)	55.8	28.4	6.3	31.8	50.8	76.2	152.4
Target Obliquity (°)	31	31.2	0	0	45	60	80
Target Young's Modulus (GPa)	207.2	5.3	69.6	207	207	207	210
Perforation	0.1205	0.8921	-0.9	-0.9	0.9	0.9	0.9
Residual Velocity (m/s)	338.3	399.3	0	0	192	585	1609
Residual Mass (g)	7.9	12	0	0	4.4	10.5	114.7

Table 1. Database statistics

Prior to being used by the neural network, all of the data were normalized to fall between the values of -0.9 and 0.9. Those values were chosen because the activation function used in the neural network is the hyperbolic tangent function. If the chosen target values "were set to the asymptotes of the sigmoid it can drive the weights to infinity, cause outlier data to produce very large gradients due to the large weights, and produce binary outputs even when incorrect" (Lawrence, Giles, & Fong, 2000).

The 2125 data points in the database were divided into 3 subsets of data: The training set consisted of 1625 data points, the validation set consisted of 181 data points, and the test set consisted of 319 data points.

The training set was used to train the ANN, the validation set was used to determine when training needed to stop, and the test set was used to determine the performance of the ANN against "new" data.

Model Selection

To provide a good estimate of the generalized validation error of the topologies tested during the model selection phase, *k*-fold cross validation was used. In *k*-fold cross validation the data is partitioned into *k* nearly equal sized folds. There are *k* ANNs that are trained on *k* - 1 folds of the data; for each ANN a different fold is left out. The fold that was left out during training is used to calculate the error of that ANN. The approximate generalized error of that ANN topology is the average of all *k* ANNs (Refaeilzadeh, Tang, and Liu, 2009). Using *k*-fold cross validation to determine the generalized error gives a better prediction of how well a given topology will perform against new information (Zhang, Hu, Patuwo, and Indro, 1999). Since *k*-fold cross validation was used during this phase, the training and validation sets were combined prior to being split into *k* folds. For this research a value of 10 was used for *k* based on the research of Gonzalex-Carrasco, Garcia-Crespo, Ruiz-Mezcua, and Lopez-Cuadrado (2011).

In an effort to test a broad range of possible topologies and ensure a good sampling for each one, 90 different topologies were run with 10 samples each to determine the best generalized topology.

The topology that was selected with the lowest generalized validation error consisted of two hidden layers having 8 neurons in the first hidden layer and 10 neurons in the second hidden layer (see figure 1).

Model Training

Before starting the training process with the selected topology, a genetic algorithm was used to provide good starting weights for the network.

During the model training process, the selected topology was trained on the training data set and the validation set was used for early stopping.

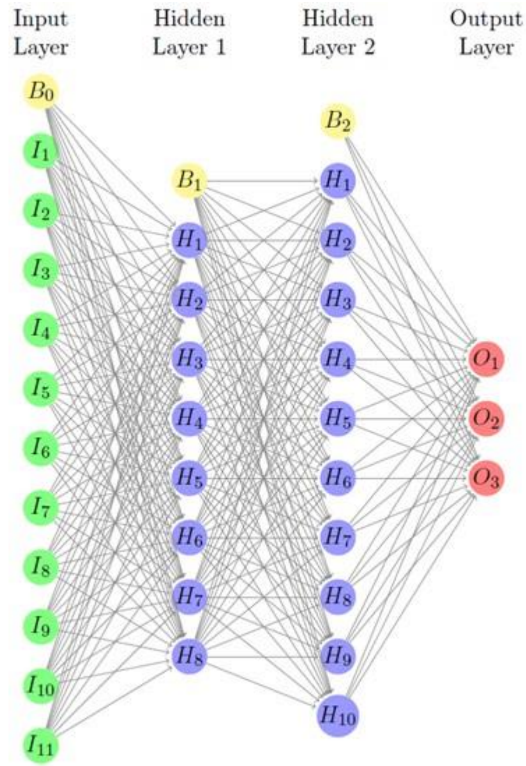

Figure 1. Neural network topology

Three strategies were used to stop the training of the ANN: Maximum iterations, minimum improvement, and generalization loss.

The first strategy used was a maximum iterations threshold. This strategy was used to limit the total number of training epochs to at most 100,000 epochs.

The second strategy used was a minimum improvement threshold with a count of epochs. This strategy considered any epoch that did not provide an improvement to the error greater than 0.00000001 as a bad epoch. The number of bad epochs allowed was set to 1000. If 1000 bad epochs occurred then training of the ANN was halted and the optimal ANN up until that point was used. If at any point prior to the bad epochs limit an improvement better than the threshold occurred, than the number of bad epochs was reset to 0.

The final strategy used was a generalization loss threshold. Generalization loss is the percent difference between the validation error (E_{va}) and the optimal error (E_{opt}) at epoch t (see equation 1). The optimal error is the best validation error up to epoch t (Prechelt, 2012).

$$GL(t) = 100 * \left(\frac{E_{va}(t)}{E_{opt}(t)} - 1\right) \quad (1)$$

A value of 7% was used for the threshold of generalization loss. While training the ANN, weights that resulted in the optimal error are saved. Upon reaching the 7% threshold training of the ANN would halt and the optimal ANN up to that point was used.

For completeness 10 samples of the chosen network topology were trained and the best one was selected to go forward to the comparison phase. Even with the use of a genetic algorithm for seeding the network weights, the subsample runs varied in their final errors from 0.226115 to 0.249743 and varied in number of epochs from 11,711 to 22,956. The generalization loss threshold strategy was used for early stopping for 6 of the 10 subsamples and the minimum improvement threshold strategy was used for the remaining 4.

3. RESULTS

For both models if the prediction of perforation would result in a non-perforation, the residual mass and residual velocity are assumed to be equal to 0. This assumption is used due to the way that most residual data are recorded. In a test event, if the projectile does not perforate the target and is embedded in the target plate, typically the velocity and mass will be recorded as a 0.

Perforation Prediction

The first output to be compared is the prediction of perforation. Figures 2 and 3 show the breakdown of perforation predictions for the ANN and the Segletes model. Out of 319 data points in the test dataset, the ANN correctly predicted the perforation outcome for 246 of the data points (77%). The Segletes model correctly predicted the perforation outcome for 199 of the data points (62%).

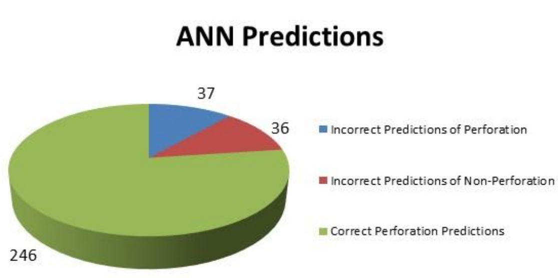

Figure 2. Perforation prediction of the ANN

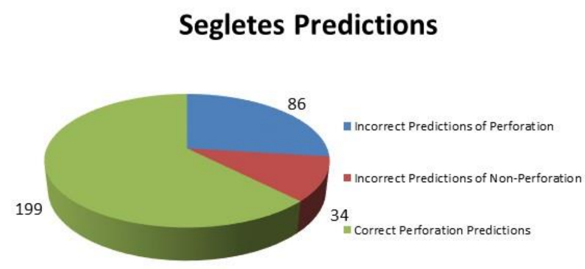

Figure 3. Perforation prediction of Segletes model

For cases where the test resulted in a non-perforation, but the model predicted perforation, the outcome is recorded as an incorrect prediction of perforation. The ANN had this occur 37 times (12%) and the Segletes model had it occur 86 times (27%).

For cases where the test resulted in a perforation, but the model predicted non-perforation, the outcome is recorded as an incorrect prediction of non-perforation. The ANN had this occur 36 times (11%) and the Segletes model had it occur 34 times (11%).

The perforation prediction statistics for all three data sets are provided in Table 2. Analyzing the results for the test set of data shows that the ANN model outperforms the Segletes model in every category except the false negative rate; the Segletes model was only about 3% better for that case.

	ANN	Segletes
Training		
MSE	0.531356	1.202289
Accuracy	75.1%	62.9%
Precision	77.7%	63.9%
False Positive	29.9%	60.1%
False Negative	21.2%	19.8%
Validation		
MSE	0.46554	1.020331
Accuracy	78.5%	68.5%
Precision	84.1%	71.3%
False Positive	24.6%	53.6%
False Negative	19.6%	17.9%
Test		
MSE	0.972121	1.289906
Accuracy	71.5%	60.2%
Precision	75.7%	62.4%
False Positive	32.6%	63.7%
False Negative	25.5%	22.3%

Table 2. Statistics for perforation prediction

Earlier versions of ANNs trained on the smaller data set of 530 data points suffered from a higher false positive rate than false negative rate. This is due to the previously mentioned high ratio of perforation to non-perforation outcomes (82%) in the old database.

With the new database ratio of perforation to non-perforation outcomes equal to 57% the ANN is able to train on a more balanced data set which leads to better overall results.

The accuracy, precision, false positive, and false negative rates for the current ANN trained on the new database, which we will call ANN_{new} and a previous ANN trained on the old database, which we will call ANN_{old} (Auten & Hammell, 2014b) are shown in table 3.

ANN_{old} was trained strictly as a classification ANN to determine perforation outcome and was trained only on the old database of 530 data points. It also had 1 less input than ANN_{new}, because the Young's modulus of the target plate material was not included in ANN_{old}.

	ANN_{old}	ANN_{new}
Training		
Accuracy	95.1%	75.1%
Precision	95.2%	77.7%
False Positive	25.4%	29.9%
False Negative	0.9%	21.2%
Validation		
Accuracy	91.1%	78.5%
Precision	90.0%	84.1%
False Positive	44.4%	24.6%
False Negative	0.0%	19.6%
Test		
Accuracy	92.5%	71.5%
Precision	91.0%	75.7%
False Positive	31.6%	32.6%
False Negative	0.0%	25.5%

Table 3. Comparison of ANN trained on old database and ANN trained on new database

At first glance it would seem that ANN_{old} performed very well at predicting perforation compared to ANN_{new}. The main problem with ANN_{old} was that it had trained to "learn" the bias in the perforation outcomes in the data. ANN_{old} was accurate 90% of time when predicting an outcome, but that was because it was mostly predicting perforation which occurred in the data 82% of the time. This can be seen by the extremely low false negative rates and the fairly significant false positive rates for ANN_{old}. Although the accuracy for ANN_{new} is less than ANN_{old}, the false positive and false negative rates are much more balanced and are manageable.

Velocity and Mass Prediction

In U.S. Army V/L simulation models velocity and mass are important for determining damage to a component that is hit by a projectile, so accurately predicting residual velocity and mass is crucial. They are also important for future applications of this ANN where those residual values will be fed back into the ANN to predict new residual values for the next armor with which the projectile interacts.

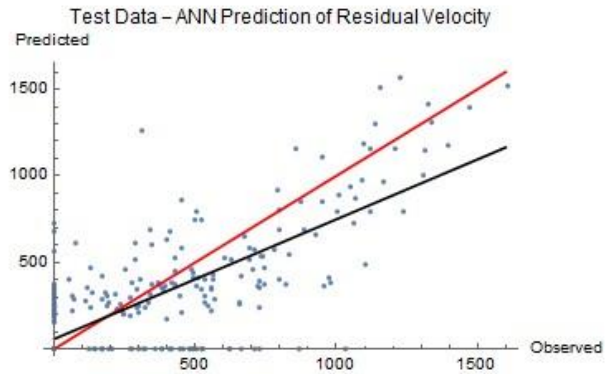

Figure 4. Correlation plot of residual velocity, observed and predicted outcomes for ANN

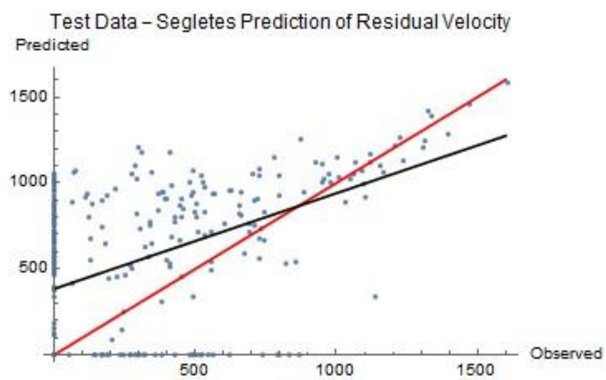

Figure 5. Correlation plot of residual velocity, actual and predicted outcomes for Segletes

Figures 4 and 5 show correlation plots of actual test outcomes against model predictions for residual velocity by the ANN and Segletes models. Figures 6 and 7 show correlation plots of actual test outcomes against model predictions for residual mass by the ANN and Segletes models. Cases of incorrect perforation predictions can be seen on the axes of the plots (where either actual or predicted equal 0). In a perfect fit scenario, all of the plotted data points would fall on a line with slope equal to 1 and y-intercept equal to 0.

Careful review of figures 4 and 5 reveal what we have already determined in the perforation prediction section of this paper. The ANN is better at predicting perforation than the Segletes model; this can be seen by the number of plotted points that fall on the axes in figure 5 compared to figure 4. There is also a better trend in figure 4 when compared to figure 5; the black line in the figures is a linear fit to the plotted points and the black line in figure 4 is much closer to the ideal red line than the one in figure 5. This demonstrates that the ANN predicts residual velocity better than the Segletes model.

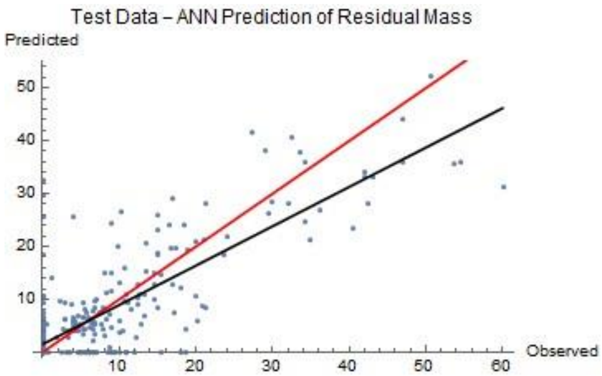

Figure 6. Correlation plot of residual mass, actual and predicted outcomes for ANN

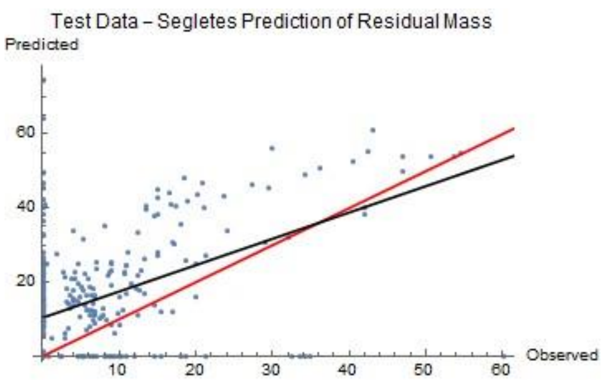

Figure 7. Correlation plot of residual mass, actual and predicted outcomes for Segletes

As with the residual velocity plots, careful review of figures 6 and 7 reveals again that the ANN is better at predicting residual mass than the Segletes model. The trend for the residual mass predictions is not as clear as it was with the residual velocity plots. The ANN has a better trend in figure 6 when compared to figure 7, but only up to about 30g on the x-axis. For values greater than 30g on the x-axis it appears that the ANN under-predicts residual mass and that Segletes over-predicts residual mass. The ANN model still has an overall better trend than the Segletes model even with the slight under-prediction after 30g.

The following plots (figures 8-11) make use of percent error, which is defined in equation 2. Percent error is undefined for any cases where the observed value is 0. That includes cases where predicted is 0 or some other value. For the cases where both predicted and observed equal 0, the percent error is recorded as 0%. For those cases where observed is 0 and predicted is greater than 0, they are not included in the graphs displaying %error. In order to ensure that the distribution of %error is not skewed, cases where predicted is 0 and observed is greater than 0 are also not included. Both of those two definitions coincide with incorrect prediction of perforation and incorrect prediction of non-perforation.

$$\%Error = 100 * \left(\frac{Observed - Predicted}{Observed}\right) \tag{2}$$

The %Error distributions for the ANN and Segletes for velocity can be seen in figures 8 and 9 respectively and mass can be seen in figures 10 and 11 respectively.

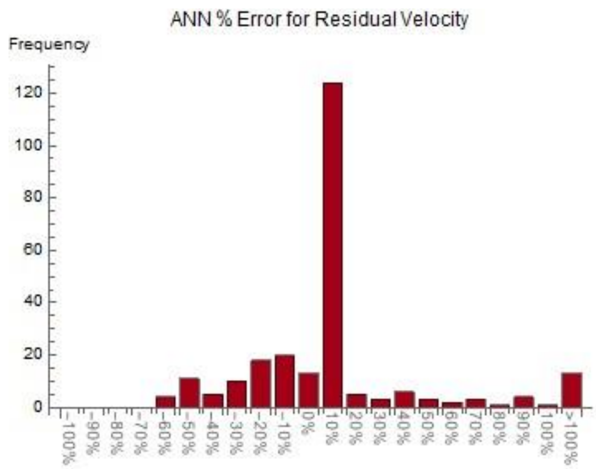

Figure 8. Distribution of %Error for ANN prediction of velocity

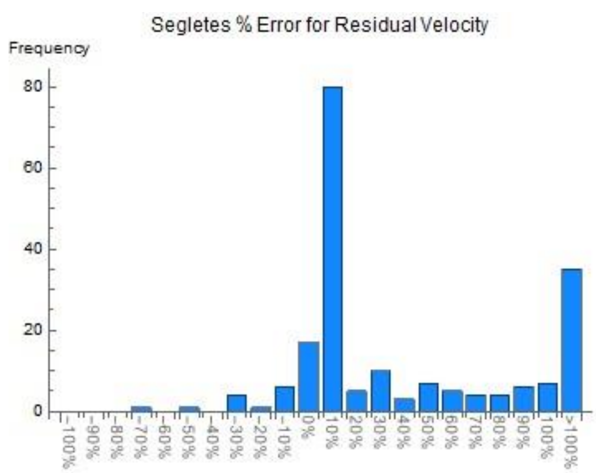

Figure 9. Distribution of %Error for Segletes prediction of velocity

Comparing the velocity plots of both models shows that although both models are fairly well centered on the 0% error bin, the Segletes model has a significant amount of predictions that had an error greater than 100%.

Comparing the mass plots of both models shows a similar result as that of the velocity plot, both models are fairly well centered on the 0% error bin, but the Segletes model has a significant amount of predictions that had an error greater than 100%.

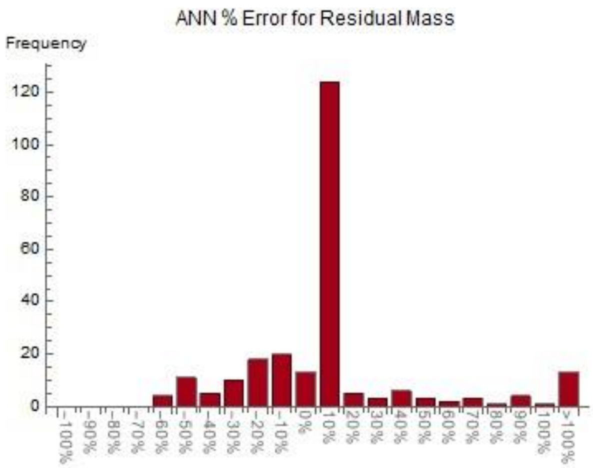

Figure 10. Distribution of %Error for ANN prediction of mass

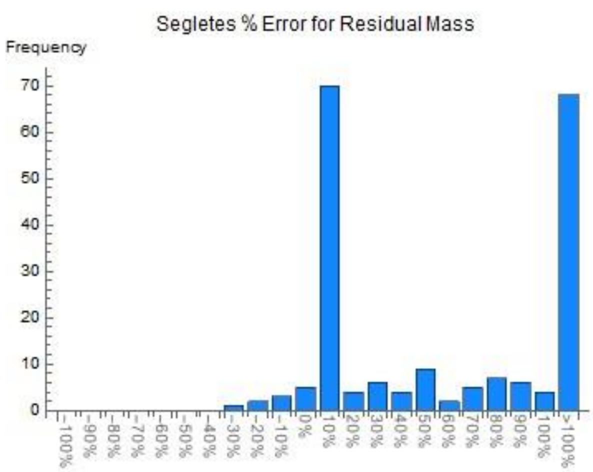

Figure 11. Distribution of %Error for Segletes prediction of mass

The previous probability distribution functions can be plotted as cumulative distribution functions with the addition of another bin that accounts for previously left out incorrect perforation predictions. That bin is placed at the far right because of the importance of correct prediction of perforation (i.e. an incorrect prediction for perforation is being treated as worse than a %error greater than 100%). The cumulative distribution function of both models for velocity is shown in figure 12 and mass is shown in figure 13.

The cumulative distribution functions of %error show a much clearer picture of the improved performance of the ANN over the Segletes model. A perfect model would jump to 100% in the first bin of <10% error, meaning that 100% of the data analyzed had an error less than 10%. So a model with a cumulative percentage rate that is always higher than another model can be thought of as always having a higher percent of data analyzed that fall in a lower percent error bin.

For both the velocity and mass distributions in figures 12 and 13, the ANN model always has a higher percentage of data analyzed with less than the given %error for that bin than the Segletes model. This shows that the ANN model is superior at predicting residual velocity and mass when compared to the Segletes model.

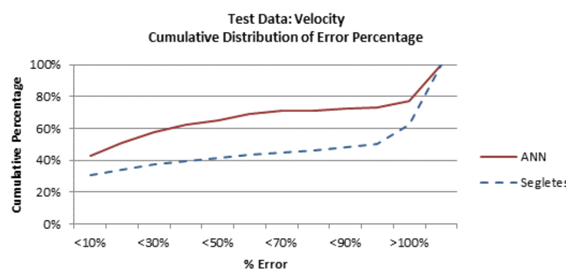

Figure 12. Cumulative distribution of %error of ANN and Segletes for velocity

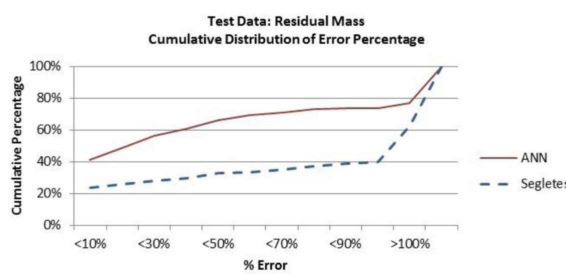

Figure 13. Cumulative distribution of %error of ANN and Segletes for mass

Model Run-Time

An analysis using a U.S. Army VL simulation can require the running of well over 1 million shotlines and on each shotline a terminal ballistics model such as Segletes or this ANN would need to be called multiple times. Given the number of times that either of these models

ISSN: 2167-1508
v8 n3667

would need to be called for an analysis, run-time is of extreme importance.
After training of the ANN is completed the calculation of its predictions is a simple mathematical equation, but the Segletes model is a numerical integration model and therefore requires more time to run.

All of the ANN runs took less than 0.1 ms to run where the Segletes model took anywhere from 0.2ms to 2.3ms (see figure 14). The average run-time (over the entire database of test data consisting of 2125 data points) for the ANN was 0.017ms and the average run-time for the Segletes model was 1.042ms. This can also be thought of as the ANN processes 58 data points per millisecond and the Segletes model can process roughly 1 data point per millisecond.

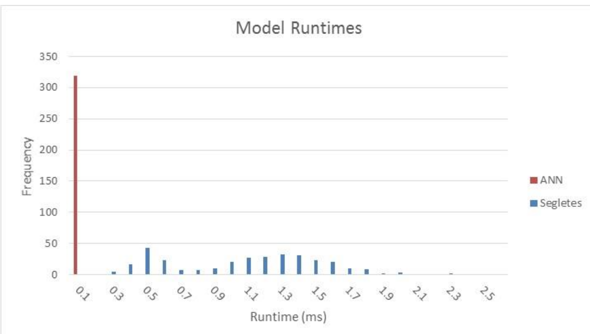

Figure 14. Distribution of run-times for both models

4. CONCLUSIONS

The results outlined above demonstrate that:

1) The ANN is superior to the Segletes model at predicting perforation of kinetic energy projectiles against homogeneous metallic armor plates.

2) The ANN is also superior at predicting the residual velocity and mass of the kinetic energy projectile after it perforates the armor plate.

3) With an average run-time that is roughly 60 times faster than the Segletes model, the ANN is far superior at processing data.

Given the three conclusions listed above, the ANN model developed as part of this research would provide the Department of Defense a model that is faster and more accurate than the best current model. This type of model would be very effective in use in larger models such as Vulnerability/Lethality simulation models and in agent-based force-on-force models where speed and accuracy are both important.

Although the ANN outperformed the Segletes model against this dataset, the Segletes model still has value for use in analyses where modeling the phenomenological processes can provide the analyst increased awareness of what is occurring during the ballistic interaction. However, if accuracy and speed is important, and increased fidelity is not required, the ANN is the better choice for the analysis.

More research is required to see how well the ANN can predict ballistic outcomes for targets that consist of more than one plate of armor. For those types of targets the ANN can be run iteratively against the plates of the target. The predicted residual values from one plate would be fed into the ANN for the next plate of armor. The final predicted residual velocity and mass coming out of the rear plate would be compared to the test results to determine overall prediction performance of the ANN.

5. REFERENCES

Auten, J. R., (2012). A Comparison of Kinetic Energy Rod Algorithm Predictions to Test Data (ARL-TR-6192). Aberdeen Proving Ground, MD: U.S. Army Research Laboratory.

Auten, J.R., & Hammell, R. (2014a). Predicting the Terminal Ballistics of Kinetic Energy Projectiles Using Artificial Neural Networks. Journal of Information Systems Applied Research, 7(1), 23-32.

Auten, J.R., & Hammell, R. (2014b). Predicting the perforation of Kinetic Energy Projectiles using artificial neural networks. 2014 IEEE Symposium on Computational Intelligence for Engineering Solutions (CIES).

Gonzalez-Carrasco, I., Garcia-Crespo, A., Ruiz-Mezcua, B, & Lopez-Cuadrado, J. L. (2011). Dealing with limited data in ballistic impact scenarios: an empirical comparison of different neural network approaches. *Applied Intelligence*, 35, 89-109.

Lawrence, S., Giles, C., & Fong, S., (2000). Natural Language Grammatical Inference with Recurrent Neural Networks. *IEEE Transaction on Knowledge and Data Engineering, IEEE Computer Society*, 12, 126-140.

Prechelt, L. (2012). Early stopping. But when? In Neural Networks: Tricks of the trade. Springer, pp. 53–67.

Refaeilzadeh, P., Tang, L., & Liu, H., (2009). Cross-Validation. In Encyclopedia of Database Systems, Liu, L. & Zsu, M., Eds. Springer US, pp. 532–538.

Segletes, S. B., (2000). An Adaptation of Walker-Anderson Model Elements into the Frank-Zook Penetration model for use in MUVES (ARL-TR-2336). Aberdeen Proving Ground, MD: U.S. Army Research Laboratory.

U.S. Department of Defense. (2008). Operation of the Defense Acquisition System. Department of Defense Instruction (DoDI) 5000.02, Washington, DC: U.S. Government Printing Office.

Zhang, G., Hu, M., Patuwo, B., and Indro, D., (1999). Artificial neural networks in bankruptcy prediction: General framework and cross validation analysis. European journal of operational research, 116(1), pp. 16–32.