ISSN: 2167-1508
v6 n2808

Predicting the Terminal Ballistics of Kinetic Energy Projectiles Using Artificial Neural Networks

John R. Auten Sr.
jauten1@students.towson.edu

Robert J. Hammell II
rhammell@towson.edu

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

Abstract

The U.S. Army requires the evaluation of new weapon and vehicle systems through the use of experimental testing and vulnerability/lethality modeling & simulation. The current modeling and simulation methods being utilized often require significant amounts of time and subject matter expertise. This means that quick results cannot be provided to address new threats encountered in theatre. Recently, there has been an increased focus on rapid results for modeling and simulation efforts that can also provide accurate results. Accurately modeling the penetration and residual properties of a ballistic threat as it progresses through a target is an extremely important part of determining the effectiveness of the threat against that target. This paper proposes the application of artificial neural networks to the prediction of the terminal ballistics of kinetic energy projectiles. By shifting the computational complexity of the problem to the fitting (regression) phase of the algorithm, the speed of the algorithm during an analysis is improved when compared to other terminal ballistic models for kinetic energy projectiles. An improvement in overall analysis time can also be realized by removing the need for input preparation by a subject matter expert prior to using the algorithm for an analysis.

Keywords: Kinetic Energy Projectiles, Terminal Ballistics, Artificial Neural Networks, Data Mining.

1. INTRODUCTION

When a U.S. Soldier takes a weapon system into the field for the first time, that Soldier wants to know that the weapon system will perform as expected. In order to ensure that the Department of Defense (DoD) acquires systems that are safe and effective; they test the system and use 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.

The focus of this research is on the development of an ANN that can predict the terminal ballistics of Kinetic Energy Projectiles (KEPs). This paper provides an overview of the proposed research and the current progress, specifically examining the issue of missing data. The paper is organized as follows: an introduction to V/L modeling is given in this section, followed by an

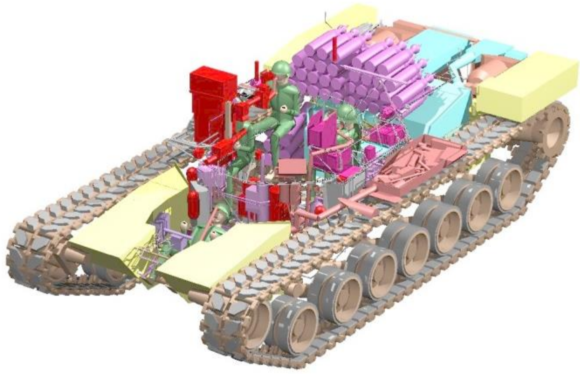

Figure 1. A CAD target model (Deitz & Ozolins, 1989)

overview of terminal ballistics in section 2. Section 3 describes the proposed modeling method and section 4 outlines the approach of this research. Section 5 presents current progress, followed by a discussion of conclusions and future research in the final section.

Vulnerability/Lethality Modeling

V/L simulation models are used to analyze the vulnerability of military systems against the lethality of weapons systems. V/L models typically consist of a Computer Aided Design (CAD) model (figure 1) of the target system, engineering definitions for the systems and sub-systems in the target, engineering inputs for the probability of component dysfunction given a hit ($P_{cd|h}$) for the target critical components, methodologies for determining system capabilities after a ballistic event, and algorithms for modeling the physical interaction of the target and the ballistic threat. This work focuses on the ballistics of the physical interaction of the threat and the target.

In V/L simulations the interaction of the target and threat is modeled as a shotline going through the target. A ballistic interaction can consist of one or many shotlines depending on the threat of interest. For example, a Shaped Charged Jet (SCJ) threat that impacts armor could generate Behind Armor Debris (BAD) which may consist of thousands of fragments, each one requiring its own shotline. Furthermore, a fragment threat could fracture upon impact and separate into several shotlines of smaller fragments.

A single interaction could require many shotlines to fully analyze the ballistic event. A typical example analysis of a target and threat could run twenty-six views or more, with each view requiring hundreds of thousands of shotlines (Moulsdale, 2012). Once all of the shotlines are tallied for a full analysis the total count can be in the hundreds of millions.

For each shotline, remaining system capability is determined based on which components are damaged. Before damage can be calculated, the model must determine if the components were hit. Determination of a hit on a component is performed by calculating how far the threat can penetrate into the target on the shotline.

An example of a shotline going through a vehicle can be seen in figure 2. The components that intersect with the shotline are considered "threatened" and are highlighted in the figure. How far along the shotline the threat can penetrate will determine which "threatened" components are actually hit. Terminal ballistics models, also known as penetration models, are used to determine how far a projectile travels on a shotline. Once the distance traveled is known, the critical components that were hit by the projectile are also known. Due to the large number of shotlines and the need for accuracy, the calculation speed and correctness of a penetration model are important.

On a particular shotline there can be many objects in the path of the projectile, so if the projectile perforates after impacting the first object on the shotline it may impact another object. For each object, a terminal ballistics model is applied to determine if the projectile will perforate the object or be defeated (Deitz, Reed Jr., Klopcic, & Walbert, 2009). The first impact event will use the initial inputs for the terminal ballistics model, and results from that impact are used as inputs to run the terminal ballistics model for subsequent impact events.

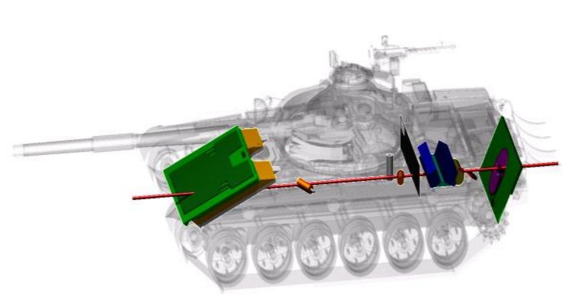

Figure 2. A shotline through a target vehicle (Dibelka, 2004)

ISSN: 2167-1508
v6 n2808

Results from the terminal ballistics model are used to determine the damage on a critical component in the target. Typically the damage to a critical component is determined using empirical models based on mass and velocity, hole size (function of projectile diameter), or energy deposited (function of mass and velocity). For each of those cases, the residual parameters of the projectile after impact are needed for determination of damage (Deitz et al., 2009). While it is important to accurately predict component perforation, it is also important to be accurate in predicting the projectile's residual parameters since they determine the damage inflicted to the target and residual penetration capability.

2. TERMINAL BALLISTIC MODELS

This research concentrates on the terminal ballistics of a particular threat type known as KEPs. KEPs are typically launched from a gun system using a sabot and can be stabilized in flight via spinning or the use of fins. They are typically made from hard and high density metals like steel, tungsten, or depleted uranium. An example KEP called an Armor Piercing Fin Stabilized Discarding Sabot-Tracer (APFSDS-T) round is shown in figure 3.

There are several models that are currently used to predict KEP penetration. Two have been chosen for discussion: the Lanz-Odermatt model, due to its simplicity and wide spread usage and the Segletes Hybrid model, due to its broad modeling capability and correctness (Auten, 2012).

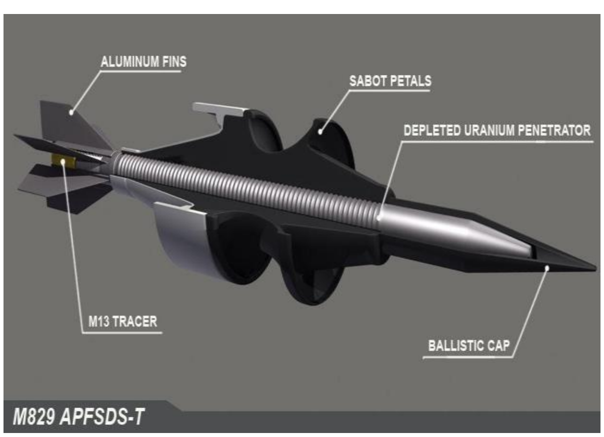

Figure 3. An APFSDS-T round ("M829", 2012)

Lanz-Odermatt

The Lanz-Odermatt model (Lanz & Odermatt, 2000) is an empirical model that is fit to test data by a Subject Matter Expert (SME). The model is very fast to run since it consists of only a few equations, but it is not a generalized model. Therefore it requires different coefficients of fit for different threats and target interactions.

Segletes Hybrid Model

The Segletes hybrid model (Segletes, 2000) is a phenomenological model built on the Bernoulli equation. It is an accurate model, but requires more run time since it uses numerical integration to solve the partial differential equations associated with it.

3. PROPOSED MODELING METHOD

Test data are available with respect to KEP penetration into various materials. However, using such data with either of the current models described above will require significant computational time or will not provide a generalized model, or both. This work investigates the use of Artificial Neural Networks (ANNs) to overcome these limitations.

Tarassenko (1998) lists five key attributes of neural networks in the book "A Guide to Neural Computing Applications":

- Learning from Experience

  Neural networks are particularly suited to problems whose solution is complex and difficult to specify, but which provide an abundance of data from which a response can be learned.

- Generalizing from Examples

  A vital attribute of any practical self-learning system is the ability to interpolate from a previous learning 'experience'. With careful design, a neural network can be trained to give the correct response to data that it has not previously encountered.

- Developing Solutions Faster with less Reliance on Subject Matter Expertise

  Neural networks learn by example and, assuming sufficient examples and an appropriate design, effective solutions can be constructed far more quickly than with

ISSN: 2167-1508
v6 n2808

traditional approaches, which are entirely reliant on experience in a particular field. However, neural networks are not wholly independent of domain expertise which can be invaluable in choosing the optimal neural network design.

- Computational Efficiency

  Training a neural network is computationally intensive, but the computational requirements of using a fully trained neural network can be modest. For very large problems, speed can be gained through parallel processing as neural networks are intrinsically parallel structures.

- Non-Linearity

  Many other processing techniques are based on the theory of linear systems. In contrast, neural networks can be trained to generate non-linear mappings, giving them an advantage for dealing with complex, real-world problems.

ANNs are a common tool for performing non-linear regression, especially when the parametric form of the function is unknown and when the number of parameters is large (Gruss & Hirsch, 2001). A specific type of ANN called a Multilayer Perceptron (MLP) has been shown to be a universal approximator, meaning it is capable of arbitrarily accurate approximation to an arbitrary mapping, if there are enough hidden neurons in the hidden layer (Gonzalez-Carrasco, Garcia-Crespo, Ruiz-Mezcua, & Lopez-Cuadrado, 2011). With the appropriate parameters, a MLP should be able to accurately approximate the desired outputs. The parameters include the inputs to the model, the topology of the MLP (to include the activation functions, number of layers, and number of neurons), the error function, the training method, and the test data.

The application of a MLP for this research was chosen based on the work of I. Gonzalez-Carrasco, et al. (2011), which found the application of MLPs to outperform Radial Basis Function (RBF) networks, Support Vector Machines (SVMs), and Recurrent Neural Networks (RNNs) for predicting perforation of steel, Depleted Uranium (DU), or Tungsten Heavy Alloy (WHA) KEPs against aluminum, steel or DU targets.

4. APPROACH

This section presents the proposed research approach. The general steps for ANN design (data preparation, determination of inputs, choice of learning method, choice of global optimization method, and use of generalization techniques) will be discussed in turn. Then, the specific ANN architecture and initial prototype used in this work will be outlined.

Data Preparation

The preparation of the data for use is an extremely important step in developing an ANN model, and is often the most time consuming. As Tarassenko (1998) states:

> *Artificial Neural Network projects are data driven, therefore there is a need to collect and analyze data as part of the design process and to train the neural network. This task is often time-consuming and the effort, resources, and time required are frequently underestimated.*

Experimental test data is inherently noisy, but hidden assumptions in the data collection methods or data processing methods could cause major differences in the data. As an example, suppose there are four reports containing experimental test data, and during the test events for all of the reports the KEP fractured into smaller pieces as it perforated the target. In report number 1, the residual mass is reported as the weight of the largest piece. In report number 2, the residual mass is reported as the weight of all of the pieces. In report 3, x-ray is used to approximate the length and diameter of the largest few pieces, and then the mass is calculated using the volume and density of the rod material. In report 4, a piece of the KEP that was embedded in the target is included in the residual mass calculation.

The above scenario produces four similar test events with four different reported results. The example given shows how important it will be to find outliers in the training data and attempt to track down the cause of the discrepancies so that they can be fixed or omitted.

In order to decrease the likelihood of poor predictions when extrapolating it is important to use training data that covers the range of all possible inputs. Figure 4 shows an example of what can happen if a region of the input space is omitted from the training data. The square

ISSN: 2167-1508
v6 n2808

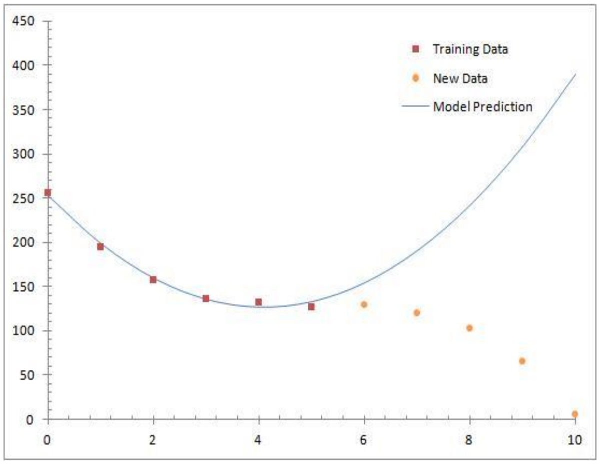

Figure 4. Example of poor extrapolation

marks are the data points that were used for the non-linear regression, the circular marks are the data points that were omitted, and the curved line shows the model predictions. The model predicts the training data very well and accurately interpolates between the data points but because of the omitted data the wrong model was used for fitting, thus leading to poor extrapolation.

The collection of experimental test data that is representative of the large space of possible input patterns and that can be used for training, testing, and validating the MLP, will be one of the more difficult tasks involved in this research (Fernández-Fdz & Zaera, 2008). Therefore, a large part of the effort for this research will be finding and documenting publicly available experimental test data for KEPs.

Determination of Inputs

Once the training data have been collected, the next step in the process of defining a MLP is the determination of inputs for the model (Walczak & Cerpa, 1999). The determination of what inputs to use is done early in the process because it drastically affects other parts of the MLP design. The number of inputs in a MLP is limited by the number of available input parameters in the problem, but it is possible that not all available input parameters should be utilized (Gonzalez-Carrasco, Garcia-Crespo, Ruiz-Mezcua, & Lopez-Cuadrado, 2008).

There is often a desire to include too many inputs in the MLP design due to two common misconceptions; (1) since they learn, they will be able to determine what input variables are important, and (2) like with expert systems, as much domain knowledge as possible should be included into the system (Walczak & Cerpa, 1999).

Determination of the input parameters is extremely important for two primary reasons. The first reason is that the required number of data points increases with the number of input parameters. The second reason is that including two inputs that are highly correlated introduces noise in the training data which can lead to a loss of generalization and could cause a non-convergence of the MLP (Kapoor, Pal, & Chakravartty, 2005).

Learning Methods

The next step in defining the MLP involves picking an appropriate learning method for the problem class being addressed (Walczak & Cerpa, 1999). The choice of learning method will determine how well the MLP will learn the patterns that it is being taught and includes the learning algorithm, error function, learning rate, and other optional methodologies. The optimization algorithms used for learning fall into two categories: direct (gradient-free) methods or gradient methods.

Direct methods use only the function values themselves to find the optima in question. Examples of direct methods include simulated annealing, perturbation methods, or genetic algorithms. The advantages of direct methods are that there is no need to derive or compute gradients and that the methods can find a global optimum. The disadvantages are that they can take too many iterations to converge to a solution and although they can come to a solution close to a global optimum, there is no guarantee that they will come to that exact solution.

Gradient methods use the gradient of the function to determine the optima in question and can be further defined as 1^{st} or 2^{nd} order. Examples of gradient methods include gradient descent, Newton method, Gauss-Newton method, and Levenberg-Marquardt method. The primary difference between a 1^{st} order and 2^{nd} order method is the required number of iterations prior to convergence and speed of calculation. 1^{st} order methods only need to calculate the 1^{st} derivative of the function which requires less calculation time, but may take a less directed approach to finding the optimum.

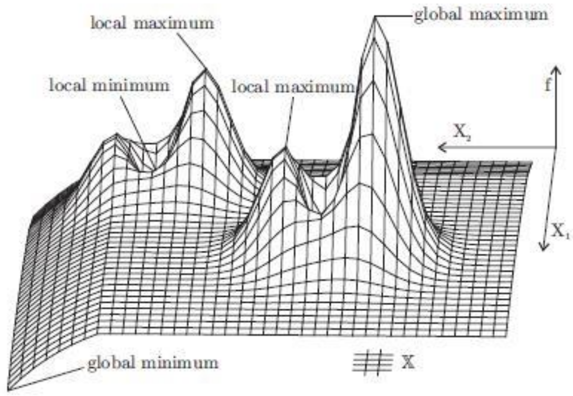

Figure 5. 3-Dimensional example of local and global optima (Weise, 2009)

2nd order methods require longer to calculate 2nd derivatives or the Hessian matrix, but take a more direct approach to finding the optimum (Snyman, 2005).

Any of the example optimization methods can be used to find a minimum of an error function, however a global minimum for the error function is not guaranteed. A hybrid of the two methods will be used for this research and will be discussed in the next section.

Methods for Global Optimization

A function can have multiple optima; figure 5 shows an example function that contains four maximums and three minimums, but there is only one global (overall) maximum and only one global minimum. An optimization function that does not guarantee the convergence to a global optimum could converge to a non-optimal solution if other methods are not used.

There are several techniques available to increase the likelihood of finding the global minimum for the error function. One technique that can be used is the method of momentum; momentum is used to resist changes to the direction of the weight changes. The main reason for using momentum is to reduce the chance of oscillating around a minimum; however, there is a slight chance that since momentum can also speed up the weight adjustments that it may skip over a small local minimum (McInerney & Dhawan, 1993). Momentum was not originally designed for finding global minimums and its probability of skipping a local minimum is small, so other techniques are better suited for this purpose.

Another technique that can be used is to sample several random potential weights for the network and start with the one that has the lowest error. The random sampling technique in no way guarantees a global minimum, but does help the learning process by allowing the network to start the learning process as close to a minimum solution as possible and could start the learning process close to a global minimum (Kapoor et al., 2005). A disadvantage of this method is that since it is truly random it is not a directed approach and is therefore inefficient when compared to directed methods.

A technique that has gained popularity is to use a hybrid approach that attempts to utilize the benefits of direct and gradient optimization methods together. Since direct methods are traditionally better equipped to find global optimum, a direct method is used first to get close to a global optimum. Direct methods however are typically inefficient in converging to the optimum solution, so the next step is to apply a gradient method to assist in the convergence.

An example of this technique is the use of Genetic Algorithms (GAs); they can be used to determine starting weights for the network prior to the learning process beginning. Like with random sampling, using a GA does not guarantee a global minimum, but does increase the likelihood of finding it since it is a directed method and is more efficient than random sampling (McInerney & Dhawan, 1993). Once a criterion has been met by the GA the learning process begins using a gradient method for the determination of the required weights to reach the global minimum of the error function. This hybrid method is what will be used for this research.

Generalization Techniques

As mentioned earlier, it is important to this research project that any model developed be a generalized solution. If non-representative data is used to train the MLP then poor extrapolation could occur. But even if representative data is used for training, if the MLP is not properly designed then it could over-predict and not provide a smooth fitting of the training data. As an example, Figure 6 shows a model that has been overfit to the training data. The diagonal line represents a good fit to the training data points, but the curved line represents a solution that could come from a MLP if overfitting occurs.

ISSN: 2167-1508
v6 n2808

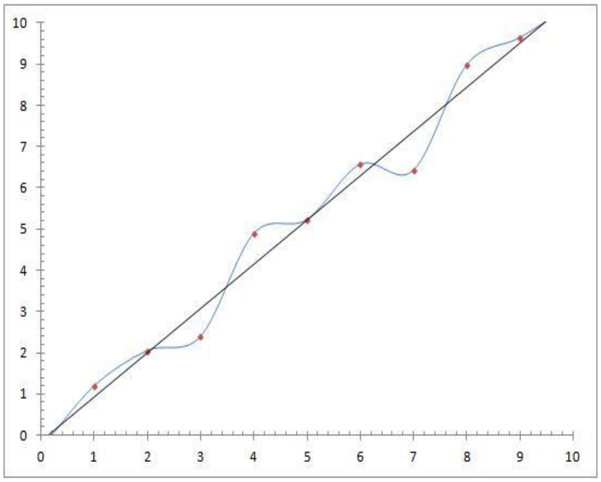

Figure 6. An example of overfitting to data

There are techniques available to increase the likelihood of producing a generalized solution and reduce the risk of overfitting. One such methodology is weight decay; it penalizes large weights in the network and causes the weights in the network to converge to smaller absolute values. Excessively large weights in the network can lead to excessive variance of the outputs from the network (Sarle, 2002). Another method for producing a more generalized model is to use early stopping. During the training phase a training set of data is used for learning as usual, but the error of a validation set of data is also tested. If the error of the validation data set begins increasing then the training is stopped early (Gonzalez-Carrasco, Garcia-Crespo, Ruiz-Mezcua, & Lopez-Cuadrado, 2011).

ANN Architecture

The work of Fernández-Fdz, Puente, and Polo (2008) used an application of ANNs that broke the prediction of residual values into a two step process. Instead of using one MLP for determining perforation and residual values, the task was broken up into a MLP for classification (perforation and non-perforation) and if perforation was predicted, a second MLP for regression of the residual values. The benefit of separating the two tasks is the reduction in complexity of the overall networks and therefore an increase in the likelihood of faster convergence.

The design for this research will follow a similar approach. The modeling of the terminal ballistics of KEPs will be broken into two sub-problems, one of classification (perforation or non-perforation) and one of regression (determination of residual parameters).

The effect that MLP complexity has on the amount of training data required can be demonstrated by using equations 1 and 2. They provide an approximation of the number of training data points required for a given network topology, or reciprocally the size limitation of a network topology due to the number of training data points (Tarassenko, 1998). In equation 1, n is the number of training data and W is the total number of network parameters (the network parameters are the weights associated with the connections between the nodes in the ANN) that must be adjusted during training.

$$W \leq n \leq 10W \tag{1}$$

$$\mathrm{W} = \sum_{\mathrm{i}=1}^{\mathrm{N}-1}(\mathrm{L_i}+1)\mathrm{L_{i+1}} \tag{2}$$

For example, for a simple 2-layer MLP with two input neurons, two hidden neurons, and one output neuron, the recommended number of training data fall between nine and ninety. For a 3-layer MLP with six input neurons, seven hidden neurons in the first hidden layer, six hidden neurons in the second hidden layer, and three output neurons, the recommended number of training data fall between one hundred eighteen (118) and one thousand one hundred eighty (1180). The more complex the MLP the more data are required for training.

Initial Prototype

Due to the simplified complexity of the problem, the first prototype will concentrate on the problem of determining perforation of a single plate of homogeneous armor. After that ANN has been developed, the next step will be to develop an ANN to determine the residual parameters for the KEP. The ANN will be applied by determining perforation for each plate along the shotline and utilizing the residual parameters for any subsequent plate along the shotline.

As more test data become available, or as gaps in data are exposed and filled using Finite Element Methods (FEM), the ANNs can be refit and refined to better model the kinematics of terminal ballistics.

5. CURRENT PROGRESS

Experimental Test Data

The experimental test data have been categorized into three main types; semi-infinite, finite, and limit. Semi-infinite test data comes

ISSN: 2167-1508
v6 n2808

from a penetration test into a material that is of such thickness that the area of deformation in front of the projectile is not expected to reach the rear face of the target. Finite test data comes from a test where the target material is of a finite thickness and under certain circumstances the projectile could perforate the target. Limit test data comes from many finite test series to determine at what velocity perforation would occur 50% of the time; this is known as the ballistic limit or v_{50}. This research is currently focused on finite test data.

There are many physical properties that are typically recorded during experimental tests, but some of the more typical ones are:

- Impact parameters such as velocity, yaw, pitch, and total yaw.
- Projectile properties such as length, diameter, density, mass, and hardness.
- Target properties such as thickness, obliquity, density, and hardness.
- Residual values such as velocity, projectile mass, and projectile length.

XML Database

The database being used for this research was designed using an XML schema. Once the schema was developed, a Java library called JAXB was used to create an object model to store the database and provide read and write access to the XML file from within a Java Swing GUI. That tool is used primarily for data entry and querying of the XML database. A Southwest Research Institute report (SwRI) (Anderson Jr., Morris, & Littlefield, 1992) was used to populate the database with its initial data set. The report was digitally scanned and then processed using Optical Character Recognition (OCR). The data from the report was cleaned up and formatted into something that was readable by a Java program. The Java program then pulled the data into the database and wrote it out in the XML format.

The seven other reports that are currently entered into the database were entered in by hand. There are currently 25 more reports of data awaiting entry into the database.

There are 1,463 records in the database that contain semi-infinite test data, 644 records that contain finite test data, and 416 records that contain limit test data.

Data Concerns

Typical problems with using large amounts of data include incorrect recording, incorrect data entry, duplication, and missing parameters.

Of the 644 records in the database, only 75 contain all 15 of currently selected variables, 451 are missing one variable, 96 are missing two variables, and 22 are missing three variables. All of the 569 that are missing values have at least one missing value that is a dependent variable.

Statistical methods have been used to expose outlier data and subject that data to scrutiny. However, further efforts are required to ensure that the data is as clean as possible. There are statistical, clustering, pattern-based, and association rules methods for outlier detection available to help with the process of cleaning the data (Maletic & Marcus, 2005).

In order to develop the initial prototype MLP for classifying the data as perforation or non-perforation, every record that is to be used for training must contain all required parameters. There is no one solution to the problem of missing data, but through a combination of intelligent replacement, imputation, or maximum likelihood methods, suitable values can be placed into the missing data locations with minimal detrimental effect to the ability of the MLP to learn the patterns in the data. The listwise and pairwise deletion methods will be avoided if possible, due to the limited availability of test data.

One method of intelligent replacement is accomplished by making the common assumption that the diameter of the KEP does not change during penetration and by using basic geometric equations. Equation 3 can be used to solve for mass (m), density (ρ), radius (r), and length (l) as long as only one of the parameters are missing.

$$\frac{m}{\rho} = \pi r^2 l \qquad (3)$$

For the remaining missing data that cannot be addressed using intelligent replacement, a determination will need to be made whether the data is missing completely at random (MCAR), missing at random (MAR), or not missing at random (NMAR). The classification of how the data is missing has important ramifications on

ISSN: 2167-1508
v6 n2808

what methods are available to fill in the data voids (Enders, 2001).

6. CONCLUSIONS

The need for an accurate and generalized terminal ballistic model for KEPs is important due to their usage in V/L models that the U.S. Army uses to evaluate the survivability of military systems. To overcome the problems inherent in current modeling and simulation methods (slow speed, need for significant subject matter expertise), this paper proposes to use artificial neural networks to produce an accurate, general model for the prediction of the terminal ballistics of kinetic energy projectiles.

The use of ANNs for regression is a well documented process in many fields. This research proposes to use the approach in a specific area in which it has not been used before. However, this work contributes in the broader context as well by examining the issue of missing data. This is a problem in almost all data-based research, and dealing with it in an unbiased way is difficult but crucial. This research will examine multiple ways of solving the issue in a practical scenario.

A literature search has been completed for publications containing KEP experimental test data and that data has been partially entered into the database. An analysis has been performed to check the correctness of the entered data and search for outliers, but data voids still pose a problem. Further analysis will be performed to address the missing data and prepare the database for usage in the ANN.

Future Research

With the database prepared, the prototype ANN will be designed, implemented, and tested. The prototype ANN will immediately have usage as a predictor of the ballistic limit for armor and will serve as the classification step in the two part process that is proposed for this terminal ballistics model.

The next phase of this research will include further development and refinement of the database and the design, development, and testing of the regression ANN that will serve as the second part of the terminal ballistics model being proposed.

7. REFERENCES

Anderson Jr., C. E., Morris, B. L., & Littlefield, D. L. (1992). *A Penetration Mechanics Database* (Technical Report No. SwRI Report 3593/001). Southwest Research Institute.

Auten Sr., John R. (2012). *A Comparison of Kinetic Energy Rod Algorithm Predictions to Test Data* (Technical Report No. ARL-TR-6192). U.S. Army Research Laboratory.

Deitz, P. H., & Ozolins, A. (1989, May). *Computer Simulations of the Abrams Live-Fire Field Testing* (Memorandum Report No. BRL-MR-3755). Ballistic Research Laboratory.

Deitz, P. H., Reed Jr., H. L., Klopcic, J. T., & Walbert, J. N. (2009). *Fundamentals Of Ground Combat System Ballistic Vulnerability/Lethality* (E. Edwards, W. Hacker, W. Kincheloe, & D. Bely, Eds.). AIAA (American Institute of Aeronautics & Astronautics).

Dibelka, R. E. (2004, June). *MUVES-S2 Configuration Management*. Presentation at the 9th Annual Joint Aircraft Survivability Program Office (JASPO) Model Users Meeting (JMUM), June 22–25.

Enders, C. K. (2001). A primer on maximum likelihood algorithms available for use with missing data. *Structural Equation Modeling, 8*(1), 128-141.

Fernández-Fdz, D., Puente, J. L., & Polo, R. Z. (2008). Prediction of the behaviour of CFRPs against high-velocity impact of solids employing an artificial neural network methodology. *Composites Part A: Applied Science and Manufacturing, 39* (6), 989–996.

Fernández-Fdz, D., & Zaera, R. (2008). A new tool based on artificial neural networks for the design of lightweight ceramic-metal armour against high-velocity impact of solids. *International Journal of Solids and Structures, 45* (25-26), 6369–6383.

Gonzalez-Carrasco, I., Garcia-Crespo, A., Ruiz-Mezcua, B., & Lopez-Cuadrado, J. L. (2008). Neural Network Application for High Speed Impacts Classification. In *Proceedings of the World Congress on Engineering* (Vol. 1).

ISSN: 2167-1508
v6 n2808

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* (1), 89–109.

Gruss, E., & Hirsch, E. (2001, May). Approximating the Ballistic Penetration Function of a Jet in a Multi-Cassette Target by the use of Neural Networks. In *Proceedings 19th International Symposium of Ballistics 7–11 May 2001* (p. 1069-1075).

Kapoor, R., Pal, D., & Chakravartty, J. (2005). Use of artificial neural networks to predict the deformation behavior of Zr-2.5 Nb-0.5 Cu. *Journal of Materials Processing Technology, 169* (2), 199–205.

Lanz, W., & Odermatt, W. (2000, April 11-14, 2000). Minimum Impact Energy For KE-Penetrators in RHA-Targets. In M. Giraud & V. Fleck (Eds.), *Proceedings of the European Forum on Ballistics of Projectiles* (p. 349-365). ISL, French-German Research Institute of Saint-Louis.

M829. (2012). In *SBWiki*. Retrieved from http://www.steelbeasts.com/sbwiki/index.php/M829.

Maletic, J. I., & Marcus, A. (2005). Data Cleansing - A Prelude to Knowledge Discovery. In O. Maimon & L. Rokach (Eds.), *The Data Mining and Knowledge Discovery Handbook* (p. 21-36). Springer.

McInerney, M., & Dhawan, A. P. (1993). Use of genetic algorithms with backpropagation in training of feedforward neural networks. In IEEE *International Conference on Neural Networks* (Vol. 1, p. 203-208). IEEE.

Moulsdale, S. (2012, June). *MUVES-S2 Study: Comparison of Vulnerability Results When Varying Grid Cells Size and Number of Views* (Technical Report No. ARL-TR-5835). U.S. Army Research Laboratory.

Sarle, W. S. (2002). *Neural Network FAQ*. Retrieved from ftp://ftp.sas.com/pub/neural/FAQ.html.

Segletes, S. B. (2000, September). *An Adaptation of Walker-Anderson Model Elements Into the Frank-Zook Penetration Model for Use in MUVES* (Technical Report No. ARL-TR-2336). U.S. Army Research Laboratory.

Snyman, J. A. (2005). *Practical Mathematical Optimization: An Introduction to Basic Optimization Theory and Classical and New Gradient-Based Algorithms* (Vol. 97; P. M. Pardalos & D. W. Hearn, Eds.). Springer.

Tarassenko, L. (1998). *A Guide to Neural Computing Applications*. New York: Arnold.

U.S. Department of Defense. (2008, December). *Operation of the Defense Acquisition System* (Department of Defense Instruction No. DoDI 5000.02). Washington, DC: Government Printing Office: U.S. Department of Defense. Retrieved from www.dtic.mil/whs/directives/corres/pdf/500002p.pdf

Walczak, S., & Cerpa, N. (1999). Heuristic principles for the design of artificial neural networks. *Information and software technology*, 41 (2), 107–117.

Weise, T. (2009). *Global Optimization Algorithms – Theory and Application*. it-weise.de (self-published): Germany. Retrieved from http://www.it-weise.de/projects/book.pdf.