Exo-atmospheric Discrimination of Thrust Termination Debris and Missile Segments

Cheryl Resch

This article explores a time-delay neural network (TDNN) for exo-atmospheric discrimination of a missile reentry vehicle (RV) from other missile parts and thrust termination debris. The TDNN is an enhanced version of a back-propagation neural network that accounts for the features in the time domain by using the rate of change of the infrared signature over several seconds as a discriminant. We used simulated infrared signatures to train and test the TDNN on 90 randomly selected scenarios. Results showed that the TDNN could identify the RV in 97% of the cases, for a leakage rate of 3%; the false alarm rate (percentage of cases for which a non-RV was identified as an RV) was 5%.

(Keywords: Target discrimination, Time-delay neural network.)

INTRODUCTION

The ability of an interceptor to discriminate between a reentry vehicle (RV) from booster parts and other debris is critical to theater ballistic missile defense. During the exo-atmospheric portion of a flight, separating and tumbling targets are especially difficult to discriminate using available techniques because all pieces follow the same trajectory. Discrimination techniques that use a ballistic coefficient (an endo-atmospheric property) cannot be applied for exo-atmospheric intercepts. Discrimination techniques that use the difference in size as a feature do not work for tumbling missile pieces. For example, if the largest piece is viewed at an aspect angle of 0° (“nose on”), and a smaller piece is viewed at an aspect angle of 90°, the smaller piece will look larger.

During separation, the missile splits into the RV (with payload), the booster, and the attitude control module (ACM). Separation debris (bolts, bands, etc.) is also generated. Thrust termination debris results when solid fuel is expelled from thrust termination ports that are used to pull the booster away from the RV during separation. The solid fuel rattles around in the fuel tank until it becomes small enough to fit through the port and, by chance, exits the port. Thrust termination debris is hot, but not burning; its exit temperature is the temperature inside the tank. The debris then cools by radiation to the exo-atmospheric environment.

This article describes a time-delay neural network (TDNN) to discriminate an RV from missile segments and thrust termination debris using data from an infrared (IR) sensor onboard the interceptor missile. The IR sensor views the threat for about 10 s, starting about 30 s before intercept. The TDNN uses data from the first 3 s of the 10-s window to discriminate the RV. The threat complex is about 100 km away from the IR sensor 30 s before intercept. At this range, even relatively large pieces are represented as single pixels on the IR image. Larger pieces cover a larger portion of the pixel, so they appear as brighter dots than do smaller pieces. Hot pieces return a larger irradiance to the sensor, so they look brighter than cooler pieces. Thus, hot and small pieces of thrust termination debris may be confused with cooler and larger missile pieces at this range. As a result, static IR images are insufficient to discriminate the RV from other missile pieces and thrust termination debris.

Temporal characteristics serve as potential discriminants since different pieces cool at varying rates, thereby causing their IR signatures to change at different rates. In particular, the payload in an RV is dense, so it cools more slowly than the empty booster tank, the ACM, or debris. Using temporal characteristics over a 3-s interval, the TDNN correctly discriminates the RV even under momentarily unfavorable aspect angle combinations. The tumble rate is generally high enough such that, over the 3-s window, the larger pieces will tend to return larger irradiance values.

We can easily incorporate new information as well as new sources of information (e.g., range data, intelligence information) by restructuring and retraining the TDNN. These factors make the TDNN an attractive choice for use in exo-atmospheric discrimination.

SIMULATED IR SIGNATURES

The model for thrust termination debris is based on a description generated by Massachusetts Institute of Technology/Lincoln Laboratory using measured data.[1,2] The debris was modeled as flat rectangular plates. Ranges for length, width, and height were determined on the basis of measured data, and values for individual models were chosen using random draws for those variables. The temperature of the debris at birth was 1500 K. It was then cooled by radiation over the entire exo-atmospheric portion of the flight (hundreds of seconds). Simulated IR signatures for thrust termination debris were generated and included in the training and test sets.

Photon Research Associates generated the IR data for the RV, booster, ACM, and separation debris. We used an aerodynamic heating program developed by Orbital Sciences Corporation[3] to predict the upper- and lower-bound temperatures of the missile during its trajectory and applied a visible and IR signature (VISIG)[4] code developed by Photon Research Associates to obtain IR signatures based on temperature histories and trajectories.

Data included IR signatures for the missile parts over a range of wavebands for 200 different combinations of dynamic parameters. These dynamic parameters, which varied for each scenario, included precession rate, coning angle, and spin rate (Fig. 1). A dual long-wave IR sensor (i.e., with two wavebands in the 7- to 11-μm range) was modeled. (A sensor of this type is currently being designed for an exo-atmospheric kill vehicle.) Sensor designers chose the long-wave region of the IR spectrum because bodies at temperatures typical of an exo-atmospheric missile emit a large portion of their total emissive energy in this region.

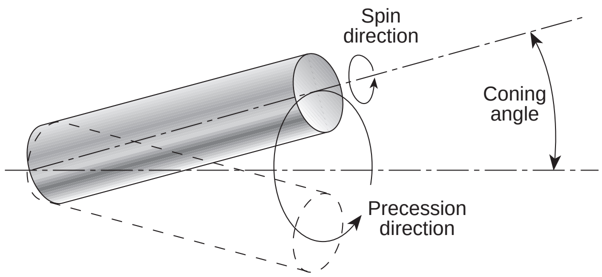

Figure 1. Schematic of dynamic parameters that describe the motion of missile pieces.

For this study, we generated 90 scenarios. A scenario is described by a random choice from the 200 combinations of dynamic parameters, a randomly chosen exo-atmospheric intercept altitude, and a combination of three binary variables (aligned RV versus RV with misalignment built in, upper- versus lower-bound temperature prediction, and one-pulse versus two-pulse trajectory).

The axis of symmetry of the misaligned RV is tilted 1° from the principal axis so that spin-related features will be present. During boost, one rocket motor pulse burns for the one-pulse trajectory and two rocket motor pulses burn for the two-pulse trajectory. (The two-pulse trajectory is longer.) Each of the eight combinations of these binary variables is represented by the same frequency in the training and test sets.

Noise was applied to the IR signature data using the following equation:

$$R_{noisy,p,w}(t) = \left[e_{cal} + e_{white}(t)\right] \times \left[R_{p,w}(t) + e_{nei}\right],$$

where

$R_{noisy,p,w}(t)$ = the noisy signal,
$R_{p,w}(t)$ = the irradiance value obtained from the VISIG code for the piece p and waveband w,
e_{cal} = calibration noise,
$e_{white}(t)$ = white noise, and
e_{nei} = noise equivalent irradiance.

The value of e_{cal} is assumed to be constant over a trajectory or scenario; for each scenario, it is randomly drawn from a Gaussian distribution from 0.9 to 1.1, with a mean of 1.0 and a variance of 0.1. The term e_{white} t) includes both electronics and photon noise, the latter caused by random fluctuations in the incident flux of photons on the detector. This noise varies for each radiance map obtained from the VISIG code; at each time step it is randomly drawn from a uniform distribution of 0.925 to 1.075. The term e_{nei} is the target irradiance at the entrance pupil of a detection system that produces a signal-to-noise ratio of 1. It is assumed to be constant over a trajectory or scenario and is randomly drawn from a uniform distribution of 0 to 1×10^{-14} W/cm^2. The model assumes that each piece is tracked from one time step to the next.

TIME-DELAY NEURAL NETWORK

A TDNN is similar to a back-propagation neural network, except that the TDNN keeps track of the data in the time domain (see the boxed insert). During each iteration, data for the current time step and the previous six time steps are used to classify each piece (e.g., debris, RV). Figure 2 shows a schematic of a TDNN. If only the black nodes (contribution of the current time step) were present, the network would be a simple back-propagation neural network. The red (time-delay) nodes represent the contribution in the time domain.

The data are gathered for 3 s in 0.3-s increments (11 time steps) starting, as noted earlier, 30 s before intercept and are then input to the neural network sequentially for each piece in the scene.

All the temporal data for the first piece are fed to the TDNN, and during each time step, the TDNN produces output values indicating which type of piece it predicts is most likely. Then all the temporal data for the second piece are fed to the TDNN, and so on. The TDNN input values i or features for a piece are the ratios of the piece's irradiance to the time-averaged irradiance values of each other piece in the scene. The following equation describes one of the input features for piece p:

$$i_{p,n,w} = \frac{R_{noisy,p,w}(t)}{\bar{R}_{noisy,n,w}} ,$$

where $\bar{R}_{noisy,n,w}$ is the mean irradiance value over the 3-s acquisition time for piece n and waveband w.

Input to the TDNN for piece p consists of one of these ratios for each other piece n in the scene for each waveband studied. If there are four pieces, there are three inputs for each waveband. For example, the input for the first piece for each waveband is $i_{1,2,w}$, $i_{1,3,w}$, and $i_{1,4,w}$. If there are two wavebands studied, there are six inputs to the TDNN for each piece.

After the data are acquired, the mean radiance values are calculated for each piece, and the sets of ratios for each piece are calculated. Using ratios rather than absolute irradiance values makes the features robust to range, time of day, and other factors than can change the overall signal to the IR sensor. For the first time step, the ratios are arranged in order of increasing magnitude. For subsequent time steps, the input positions remain constant, even if the order of the magnitudes changes. For example, if the first piece is being examined, and the inputs to the TDNN for the first time step are in the order $i_{1,2,w}$, $i_{1,4,w}$, and $i_{1,3,w}$, these input positions remain constant, even if the fourth piece eventually becomes brighter than the third.

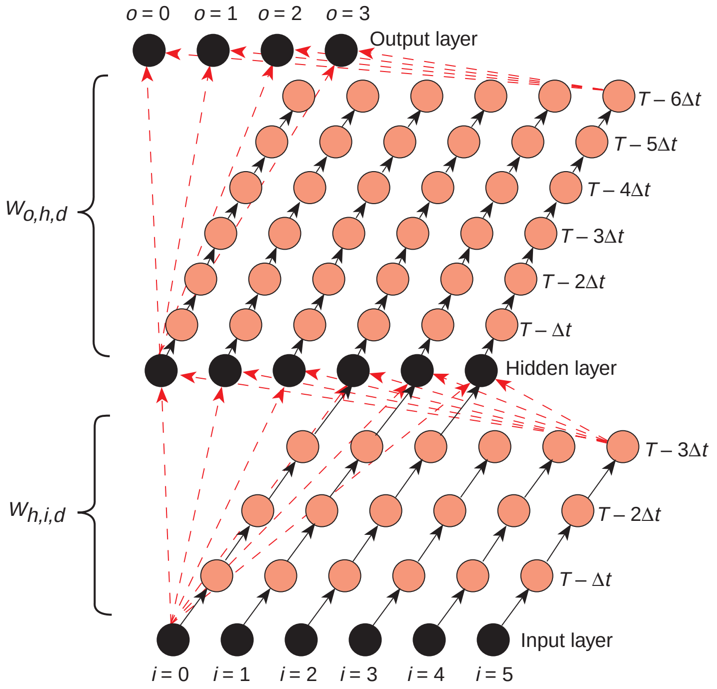

Figure 2. Schematic of the TDNN. Black circles = contribution of the current time step; red circles = time-delay nodes; dashed arrows = weight updated by back propagation; solid arrows = time-delay connection. For $W_{o,h,d}$, all nodes are fully connected to the output layer; for $W_{h,i,d}$, all nodes are fully connected to the hidden layer.

Some of the pieces are orders of magnitude different in irradiance than others, with the ratios ranging from 1000 to 1×10^{-3}. To help the neural network distinguish such large variations, the logarithm of each ratio is taken before it is input to the TDNN.

Unlike conventional neural networks, which input static patterns, a TDNN can deal with patterns in the time domain.[5,6,7] As discussed previously, temporal characteristics, in addition to size and temperature, are used to discriminate the RV. The TDNN accounts for temporal changes in the data using time-delay nodes.

After a set of inputs is introduced to and fed forward through the neural network, they are sent to time-delay nodes ($t - \Delta t$) for use during the next iteration. After subsequent iterations, information in the time-delay nodes is sent to subsequent time-delay nodes, for example, from ($t - \Delta t$) to ($t - 2\Delta t$).

NEURAL NETWORKS

Back-propagation neural networks are the most widely used of the neural network paradigms. They typically have three or more layers of nodes, which are connected by weights W. When the network is given an input, the updating of activation values propagates forward from the input layer of nodes, through each internal layer, to the output layer of nodes through the weights. The output nodes are the neural network's response. During training, the neural network corrects itself by comparing the output node value to the desired output node value. This comparison is propagated backwards through each internal layer to the input layer. During back-propagation, the weight connections between the layers are changed so that the output node value will be closer to the desired output node value during the next iteration.

A time-delay neural network (TDNN) is similar to a back-propagation neural network, except that it accounts for the features in the time domain or the order in which the features are input. (See Fig. 2 in the text for a schematic of a TDNN.)

After a set of inputs is introduced and fed forward through the neural network, the input is sent to a time-delay node ($t - \Delta t$) for use during the next iteration. After subsequent iterations, information in the time-delay nodes is sent to subsequent time-delay nodes, for example, from ($t - \Delta t$) to ($t - 2\Delta t$).

The following equations describe how the input information is fed forward through the TDNN to the output nodes. The activation level of input node i, $a_i(t)$, is the value input to the TDNN. The value then sent to each hidden node h is the sum of the products of the input values and the weight connections,

$$S_h(t)=\sum_{i}\sum_{d=0}^{D} W_{h,i,d}a_i[t-(\Delta t)d],$$

where D is the total number of time delays, d is the index of the time delay, Δt is the time-delay time increment, and $W_{h,i,d}$ is the value of the weight from input node i to hidden node h at time increment d. The activation level for (output from) the hidden node h is

$$a_h(t)=f\left[S_h(t)\right],$$

where $f(x)$ is the symmetric sigmoid function. The value sent to each output node o is then the sum of the products of the hidden node activation levels and weight connections

$$S_o(t)=\sum_{h}\sum_{d=0}^{D} W_{o,h,d}a_h[t-(\Delta t)d],$$

where $W_{o,h,d}$ is the value of the weight connection from hidden node h to output node o at time increment d. The activation level of output node o (the final output) is

$$a_o(t)=f\left[S_o(t)\right].$$

During training, the weights are adjusted using back-propagation of the error values. For an output node, the error value is

$$\delta_o(t)=\left[t_o(t)-a_o(t)\right]f'\left[S_o(t)\right],$$

where $t_o(t)$ is the target value for the output node (1 for "yes" or 0 for "no" in this study), and $f'(x)$ is the derivative of the symmetric sigmoid function. For a hidden node, the error value is

$$\delta_h(t)=\left[\sum_{o}\sum_{d=0}^{D}\delta_o(t)W_{o,h,d}\right]f'\left[S_h(t)\right].$$

The adjustment of the connection weights is done using gradient descent:

$$\Delta W_{h,i,d}=-\eta\left(\frac{\partial\delta_h}{\partial W_{h.i,d}}\right),$$

where η is a learning coefficient between "0" and "1". By evaluating this expression and adding the momentum term to speed convergence, the following equation results for adjusting weights:

$$\Delta W_{h,i,d}(t)=\eta\delta_h(t)a_i\left[t-(\Delta t)d\right]+\alpha\Delta W_{h,i,d}(t-\Delta t),$$

where α is the momentum term between "0" and "1". The momentum term increases the weight change without causing it to oscillate wildly because it changes the weight in the direction of the long-term trend. Training does not begin until all the time-delay nodes are filled (i.e., the seventh time step).

Each node in the input layer has three time-delay nodes, each with its own weight connection to the hidden layer. The nodes in the hidden layer have six time delays apiece. Thus, during each iteration, the neural network output is determined by information fed forward from the current time plus the last six time steps.

We split the problem of discriminating the RV from thrust termination debris and other missile segments into two tasks, each with its own TDNN. Infrared radiance for a waveband is found by integrating Planck's equation, which is a function of temperature, over the waveband of interest.

Since the thrust termination debris is much hotter than the missile pieces, its radiance relative to that of the missile pieces will change from one waveband to the next.[1] This property is used by the first TDNN, or hot debris TDNN, to separate the hot thrust termination debris from the missile pieces. The second TDNN, or missile piece identification TDNN, examines the remaining pieces and identifies what type of missile piece each one is.

The hot debris TDNN has one output node, which returns a value for every time step: ideally, "1" if the piece is hot debris and "0" if it is not. A TDNN could be set up to take any number of pieces as input. For this study, we took five pieces of thrust termination debris and four pieces that were not, for a total of nine pieces. Since we have nine pieces, we have eight ratios for each piece for each waveband studied. Two wavebands were studied, so there were 16 inputs to the hot debris TDNN.

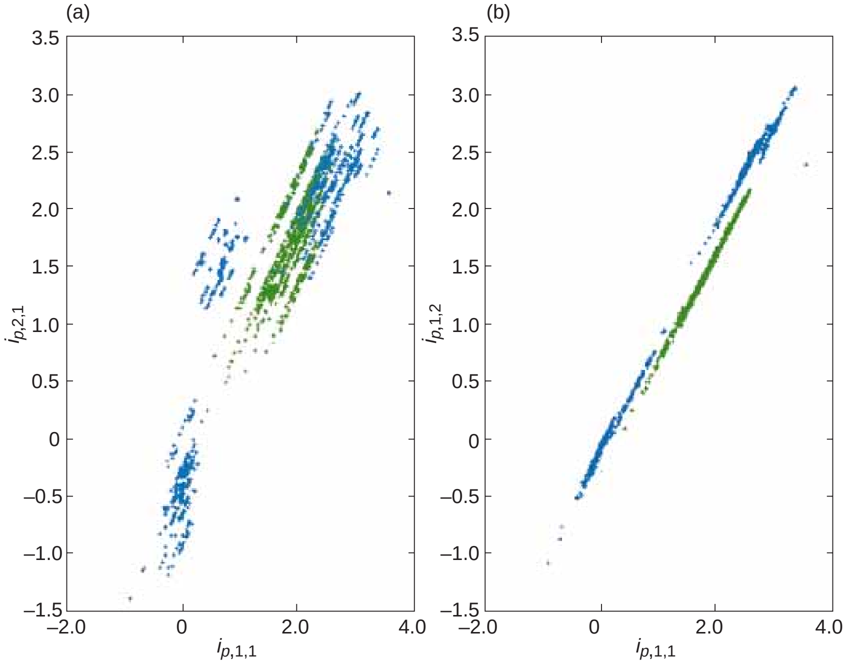

Figure 3. Hot debris TDNN input data for (a) two ratios in the same waveband and (b) one ratio in each waveband. (Missile pieces in blue; thrust termination debris in green.)

The missile piece identification TDNN has four output nodes. One output node represents whether the piece in question is an RV; i.e., output for this node should be "1" if the piece is an RV and "0" if it is not. The second output node represents whether the piece in question is a booster tank; again, the output for this node should be "1" if the piece is a booster and "0" if it is not. The third and fourth output nodes represent the ACM and separation debris, respectively. Since four missile pieces and two wavebands are modeled, there will be six inputs to the missile piece identification TDNN.

The TDNN was trained and tested for 90 different scenarios, resulting in about 5000 data points. Specifically, we used "leave-one-out" training[8]; that is, the TDNN was trained using all but 1 of the available sets of 90 scenarios. It was then tested on the left-out set. Next, the TDNN was trained again, but with a different scenario left out. The training and testing process was repeated so that each scenario was left out once. This technique gives the best estimate of the true performance of a neural network when a limited amount of data is available.

RESULTS

We first analyzed the performance of the hot debris TDNN. Figure 3a shows hot debris TDNN input for two ratios in the same waveband, $i_{p,1,1}$ and $i_{p,2,1}$, plotted against each other. The data for the hot debris and the missile pieces overlap; they cannot be distinguished using data from only one waveband. Figure 3b shows the data for one waveband input value $i_{p,1,1}$ versus input value $i_{p,1,2}$ for the other waveband. The thrust termination debris is clearly separated from the missile pieces in Fig. 3b, indicating that the change in radiance of the thrust termination debris relative to the missile pieces from one waveband to the next can be used to discriminate between them.

Figure 4 (black curve) shows the percentage of thrust termination debris test cases input to the hot debris TDNN that resulted in an output value above 0.5 as a function of time step. This debris is correctly identified in more than 95% of the cases. The TDNN also did not misclassify any missile pieces, RV, or ACM as thrust termination debris.

Figure 4 (red curve) also shows the percentage of booster input cases that incorrectly resulted in an output value above 0.5. At later time steps, less than 5% of booster inputs are incorrectly identified as thrust termination debris. Thrust termination debris is very hot, and thus very bright in the infrared. The booster is the brightest of the missile pieces because it is the biggest; therefore, the booster is the missile piece most likely to be confused with thrust termination debris.

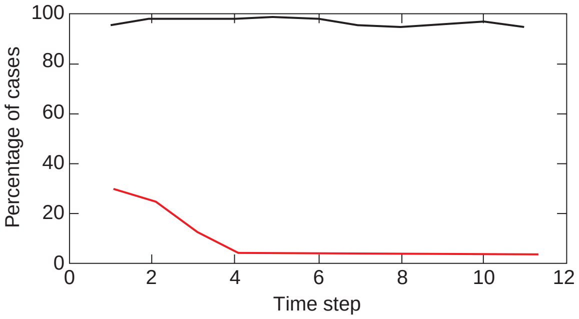

Figure 4. Percentage of test cases for which TDNN output is above 0.5 as a function of time step for thrust termination debris (black curve) and booster parts (red curve).

The TDNN can distinguish hot pieces from big pieces, even though they are represented by single pixels, because information in two wavebands separates pieces with different temperatures. Temporal characteristics are not necessary for discrimination. Hence, a simple three-layer back-propagation neural network that does not consider temporal characteristics was created that yielded performance similar to that of the TDNN. Future implementations, therefore, could use a back-propagation neural network for this portion of the problem.

Next, we used the missile piece identification TDNN to classify each missile piece. Figure 5 shows the first three ratios ($i_{p,1,1}$, $i_{p,2,1}$, and $i_{p,3,1}$) used as input plotted against each other. All three are for the same waveband. The RV and the separation debris are clearly distinguishable from the other pieces. The ACM and the booster show considerable overlap. Figure 4 indicates that this TDNN should be able to discriminate the RV and the debris from the other pieces, but will probably have trouble discriminating the booster from the ACM.

Figure 6a shows TDNN output for the RV data in the test cases. The red curve indicates the percentage of the 90 RV test cases input to the TDNN for which the RV node had the highest output as a function of time step. Ideally, this fraction should be 1, indicating that all RV test cases were identified as RVs by the TDNN. The results indicate about 97% of the RVs were identified as such. At later time steps, about 3% of the 90 RV test cases are identified as ACMs. This represents the leakage rate, or percentage of RVs that are not identified as RVs.

Figure 6b shows the TDNN output for the booster data in the test cases. Ideally, the line representing the booster node should be 1. The figure indicates that at later time steps, the ACM and booster are conflated by the TDNN. This result was to be expected given that Fig. 5 indicates that the ACM and booster feature spaces overlap significantly. However, for the test cases studied, booster data were never misidentified as RV data by the TDNN; there were no booster false alarms.

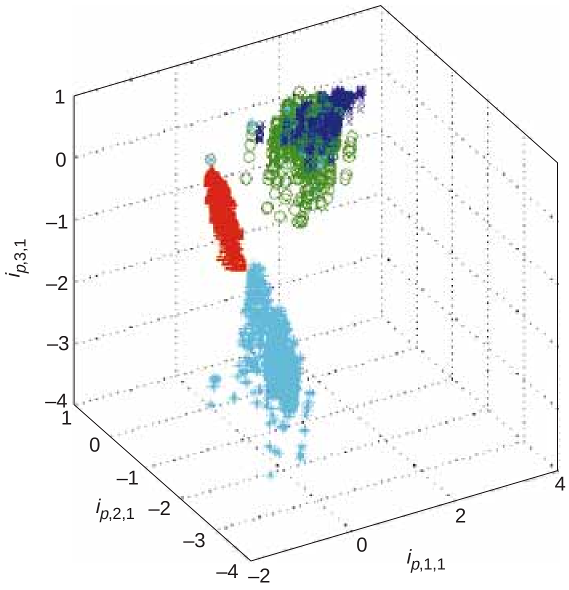

Figure 5. Input to missile piece identification TDNN for three ratios in the same waveband (booster test cases, purple; ACM, green; RV, red; debris, blue).

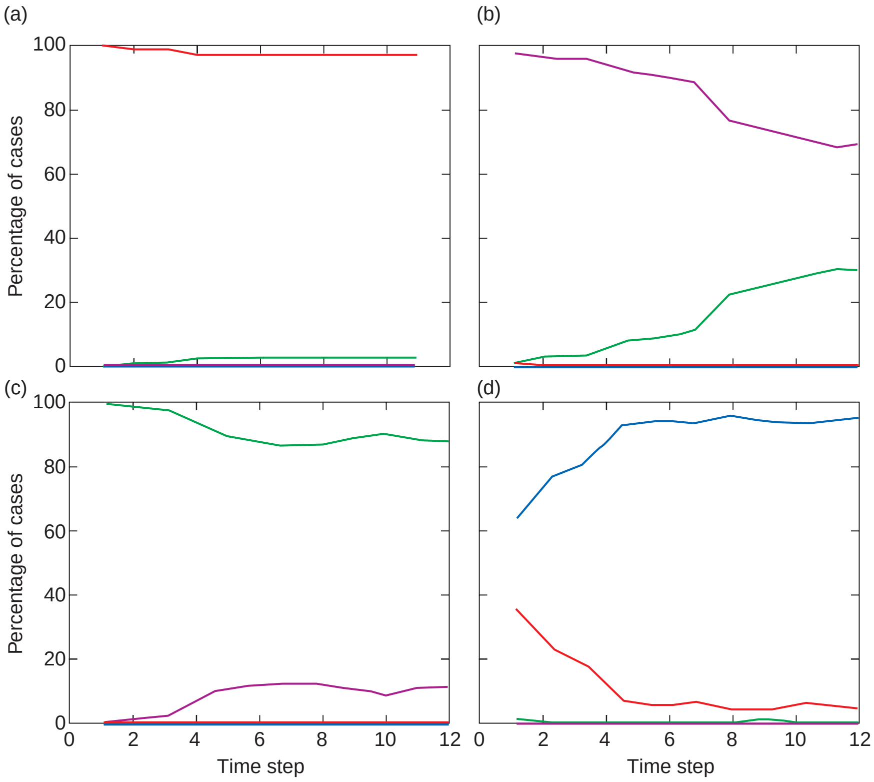

Figure 6. TDNN output (RV node, red; booster node, purple; ACM node, green; debris node, blue) for (a) RV, (b) booster, (c) ACM, and (d) debris test cases: percentage of cases for which output nodes have the highest output.

Figure 6c shows the TDNN output for the ACM data. Again, at later time steps, the ACM and booster are conflated by the TDNN. ACM data were never identified as RV data by the TDNN; there were no ACM false alarms. Figure 6d shows the TDNN output for missile debris data. At early time steps, about 35% of the debris input is identified as an RV; this drops to about 5% at later time steps, corresponding to a false alarm rate of 5% for missile debris.

When ratios for incorrectly classified thrust termination debris were input to the missile piece identification TDNN, the output nodes with the highest output values were the ACM and booster nodes. The output values were generally below 0.5. Since they are not classified as RVs, thrust termination debris that leaked from the first TDNN does not present a problem.

To determine whether temporal variations are necessary for discrimination, we trained and tested a back-propagation neural network using the same data input to the missile piece identification TDNN. For RV test cases, none of the four output nodes had a significantly higher output value than the others. The highest output value was generally below 0.5. When the TDNN is used for RV input data, the output value of the RV output node is 0.7 or above. Thus, the temporal information enhances the ability to distinguish the RV.

CONCLUSIONS

The hot debris TDNN could identify over 95% of the thrust termination debris pieces. This corresponds to a leakage rate of 5%. No RVs, ACMs, or missile separation debris were incorrectly identified as thrust termination debris. About 5% of the booster input was identified as thrust termination debris. Since the ultimate goal is to identify the RVs, eliminating the booster from consideration early is acceptable. As discussed previously, the change in radiance from one waveband to another of the hot debris versus missile pieces is used to discriminate them. Since temporal variations are not used, a simple back-propagation neural network could be applied for this portion of the problem.

The missile piece identification TDNN could identify 97% of the RVs as RVs, with a leakage rate of 3% and false alarm rate of 5%. Boosters and ACMs were confused, but again, the goal is to identify RVs, so this is not a problem.

This approach was successful because it accounts for the long-term temporal changes in the IR signature of the missile pieces as they cool. The TDNN is appropriate for use with time series data having discriminating characteristics present over a span of several time steps.

The use of long-term temporal changes also makes the approach robust to the aspect angle of the missile piece being viewed. The use of ratios as features makes the approach robust to time to go before intercept, time of day, and other factors that change the overall IR signature. The TDNN shows promise as a tool to discriminate RVs from other missile pieces and thrust termination debris in the exo-atmosphere. Further tests should be based on actual flight data.

REFERENCES

[1] Resch, C. L., *Preliminary Results of Thrust Termination Debris Modeling and Discrimination,* RSI-97-021, JHU/APL, Laurel, MD (1997).

[2] Bernstein, M. D., Meins, C. K., Rennich, P. K., and Wanner, C. H., *Solid-Fuel Debris Model Description,* Massachusetts Institute of Technology Lincoln Laboratory, Lexington, MA (1997).

[3] Montenegro, E. S., *Optical Signatures Code,* 61AN24-0064, Teledyne-Brown Engineering, Huntsville, AL (1996).

[4] Filice, J., *VISIG Visualizer Users Manual,* Photon Research Associates, La Jolla, CA (1994).

[5] Resch, C. L., *Summary of Exo-Atmospheric Discrimination Algorithm Development,* AM-94-E207, JHU/APL, Laurel, MD (1994).

[6] Lin, D., Dayhoff, J. E., and Ligomenides, P. A., “Trajectory Recognition with a Time-Delay Neural Network,” in *Proc. International Joint Conference on Neural Networks,* pp. III-197–III-202, Baltimore, MD (1992).

[7] Waibel, A., Land, K. J., and Hinton, G. E., “A Time-Delay Neural Network Architecture for Isolated Word Recognition,” *Neural Networks* 3(1), 23–43 (1990).

[8] Weiss, S. M., and Kulikowski, C. A., *Computer Systems That Learn,* Morgan Kaufmann Publishers, San Mateo, CA, pp. 26–39 (1991).

THE AUTHOR

CHERYL RESCH received a B.S. and M.S. in mechanical engineering from the University of Maryland in 1986 and 1988, respectively. She joined APL in 1988 and is currently a senior engineer in the Signal and Information Processing Group of the Research and Technology Development Center. Ms. Resch worked in the Engineering Group of the Aeronautics Department until 1996, where she focused on modeling aerodynamic heating, ablation, and IR signatures of missiles. She has also worked on nondestructive evaluation of cylinder faults in internal combustion engines and reliability models of aircraft engines. Her e-mail address is cheryl.resch@jhuapl.edu.