Space object classification using deep neural networks

Conference Paper · March 2018

DOI: 10.1109/AERO.2018.8396567

Space Object Classification using Deep Neural Networks

Bin Jia
Intelligent Fusion Technology, Inc.
20271 Goldenrod Lane, Suite 2066
Germantown, MD 20876
bin.jia@intefusiontech.com

Khanh D. Pham
Space Vehicles Directorate
Air Force Research Lab
Kirtland AFB, NM 87117
khanh.pham.1@us.af.mil

Erik Blasch
Air Force Office of Scientific Research
Air Force Research Lab
Arlington, VA 22203
erik.blasch.1@us.af.mil

Zhonghai Wang, Dan Shen, Genshe Chen
Intelligent Fusion Technology, Inc.
20271 Goldenrod Lane, Suite 2066
Germantown, MD 20876
{zwang, dshen, gchen}@intfusiontech.com

Abstract—**Space object classification is desired for space situational awareness to be able to discern resident space object (RSO) characteristics, behaviors, and perspective changes. Due to the limited sensing resources and observations, it is challenging for space object classification to be responsive to unfolding and unexpected events. Many machine learning algorithms are already used to classify space objects based on various sensor observations from radar and telescope. In this paper, the use of deep neural networks (DNN) is proposed to classify space objects due to DNN robust performance in many classification tasks, such as face recognition and object recognition. This paper explores DNN using light curve data. Conventional classification algorithms, such as k nearest neighbor (*k*-NN), are implemented and compared to the proposed DNN based classification algorithms, including the popular convolutional neural network (CNN) and the recurrent neural network (RNN), in terms of accuracy. Inherent advantages and disadvantages of the deep neural network based classification algorithms are summarized and the potential for future space object classification tasks is analyzed and postulated.**

Table of Contents

1. Introduction 1
2. *K*-Nearest Neighbor with Dynamic Time Warping 2
3. Deep Neural Networks 2
4. Experiments and Testing 4
5. Summary 6
References 6
Biographies 7

1. Introduction

Space situational awareness includes monitoring of the space environment and resident space objects [1]. One of the critical tasks is to classify space objects according to their properties. Unfortunately, the information of space objects available is often limited. Typically, the visual magnitude and the radar cross section (RCS) of space objects can be obtained via optical and radar sensor, respectively. The light curve data, which includes a sequence of visual magnitude, and the RCS hence provide a way to classify space objects.

Zhang *et al*., provide an example of resident space object (RSO) classification using hyperspectral data [2] and Payne *et al*., [3] use illumination signatures for RSO classification. Many machine learning (ML) classification algorithms can be applied, such as the hierarchical clustering for the unsupervised learning and support vector machine (SVM) for supervised learning. For both kinds of learning algorithms, a data sequence can be directly used. When the length of the time-series data is long, dimension reduction algorithms can be applied [4-7] such as the down-sampling method [4]. Other methods utilizing approximating straight lines [6] or preserving the salient points [7] are also available. Besides directly using the raw or sampled data sequence in the time domain as the feature vector in the learning algorithms, the representation in a transformed domain could also be used, such as the discrete Fourier transformation (DFT) [8] and the discrete wavelet transformation (DWT) [9].

When the features of the time-series data are available, distance or similarity measures are often critical to many time-series classification algorithms. Using similarity measures can be roughly categorized as the whole sequence matching or subsequence matching. For the *whole sequence matching*, the most popular approach is to evaluate the Euclidean distance between two time series based on the transformed representations, such as using the DFT/DWT coefficients [9, 11]. However, the Euclidean distance is not always a suitable distance measure and the dynamic time warping (DTW) technique [10] is often used in applications. Based on various distance and similarity measures; further classification, indexing, and motif discovery can be processed, where more details can be found in [12].

Classical time-series classification algorithms, such as *k* nearest neighbor (k-NN), have been widely used. Due to the superb performance of the deep neural networks (DNNs) in various classification tasks, such as object detection and recognition in video, we are interested in the investigation of the space object classification using time-series data with a DNN. There are many different neural network architectures. In this paper, we use two contemporary neural

978-1-5386-2014-4/18/$31.00 ©2018 IEEE

network architectures: the convolutional neural network (CNN) and the recurrent neural network (RNN) as comparisons. Both network architectures are good at time-series classification.

The main contribution of this paper is to investigate the space object classification using a DNN. The remainder of the paper is organized as follows. Section 2 introduces the *k*-NN with dynamic time warping algorithm. Section 3 introduces DNNs, including the convolutional neural network (CNN) and the recurrent neural network (RNN). Numerical simulations and results are shown in Section 4. Section 5 provides the concluding remarks.

2. *K*-NEAREST NEIGHBOR WITH DYNAMIC TIME WARPING

k-nearest neighbor (*k*-NN) is one of the classical classification algorithms widely used in time series classification. Dynamic time warping (DTW) is a classical algorithm to measure the similarity of two time series data sets which have different sampling intervals.

Dynamic Time Warping

The dynamic time warping algorithm uses the *warp path* distance to evaluate the distance between two time series. Assume there is a desire to calculate the similarity between two time series, *X* and *Y*. The length for *X* and *Y* is denoted by $|X|$ and $|Y|$, respectively. The *warp path* is then defined as

$$W = w_1, w_2, \cdots, w_k, \quad \max\left(|X|,|Y|\right) \le k \le |X| + |Y|. \quad (1)$$

Note that each $w_m = (i, j)$, $1 \le m \le k$ corresponds to two coordinates, *i* and *j*. *i* and *j* are the index of the time series *X* and *Y*, respectively. In addition, $w_1 = (1,1)$ and $w_k = (k,k)$. Moreover, the index *i* and *j* should monotonic increase with increasing *k*.

The warp path distance is then given by

$$D(i,j) = Dist(i,j) + \min\left(D(i-1,j), D(i,j-1), D(i-1,j-1)\right). \quad (2)$$

The DTW algorithm can find a good match (e.g., smallest distance) between two time-series. Hence, *k*-NN is often integrated with DTW algorithm in time-series classification problem as the benchmark.

Recurrence Plot as a Textual Feature

The feature vector using all data points in the data sequence often fails to capture the essential structure or motif of the data sequence, which may cause the low accuracy of the classification. The texture features of the data sequence have been used in time series classification using conventional machine learning algorithm in [13]. For some time series, which don't have representative features in the time domain, the texture feature can be used to solve the classification problem [12]. To extract the texture feature, first, one obtains the recurrence plots (RP). The recurrence plot is given by

$$R_{i,j} = \Theta\left(\varepsilon - \left\|\vec{x}_i - \vec{x}_j\right\|\right), \quad i, j = 1, \cdots, N \quad (3)$$

where N is the number of data points. *i* and *j* are indexes. ε is the threshold for closeness. $\Theta(\cdot)$ is the Heaviside step function. Note that the recurrence plot obtained by Eq. (3) is a binary image. The threshold in real applications is difficult to choose. In addition, the binary image is lossy as some information fidelity between 0 and 1 is discarded. To improve upon the binary image, the threshold is removed. In this paper, the recurrence plot is used as the alternative input of the neural network for classification.

3. DEEP NEURAL NETWORKS

Convolutional Neural Network

A Convolutional neural network (CNN) is one of the representative network architectures of the deep neural networks. CNN has achieved great success in image processing and computer vision. One of the advantages of CNN is that no handcrafted feature extraction is necessary. CNN classifier accepts the raw data input directly and uses a fully connected architecture, which leads to huge number of parameters for the neural network. Then training process is hence hard and computationally intensive. Two main techniques, local connectivity and shared weights, are applied to overcome the problem. CNN enforces the *local connectivity* pattern between neurons and adjacent layers. The *shared weights* mean some connections utilize the same parameter set (weight and bias). Hence, the computational complexity can be reduced greatly. In addition, the convolution layer is used in CNN, involving convolution operations, which is frequently used in conventional image processing. Different features can be extracted after different convolution operations. To use CNN to classify light curves, different layers should be used. Note that CNN can work for either one dimensional or two-dimensional signals. In this paper, the original light curve data (one-dimensional signal) and corresponding recurrence plot (two-dimensional signal) are used. A typical measure is cross-entropy.

The cross-entropy loss is used as the loss function to be minimized. It is given by [14]

$$L(\theta) = -\frac{1}{N}\sum_{i=1}^{N}\left(y\log\tilde{y} + (1-y)\log(1-\tilde{y})\right), \quad (4)$$

where N is the number of training samples. y and $\tilde{y}$ are the output of the neural network (predicted category) and label of samples (true category), respectively.

The adam optimizer is used to solve the y and $\tilde{y}$ parameters efficiently.

The structure of CNN used in this paper for one dimensional and two dimensional signals is shown in Figure 1 and 2, respectively. The number of layers and filters/kernels is listed in Figure 1 and Figure 2. More details of the meaning of different layers can be find in [15]. Note that batch normalization layer is described in [16]. It allows us use higher learning rates. Softmax layer uses softmax function, which is given by

$$\sigma(\mathbf{z})_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}} . \tag{5}$$

where $\mathbf{z}$ is a K dimensional vector. The subscript denotes the index.

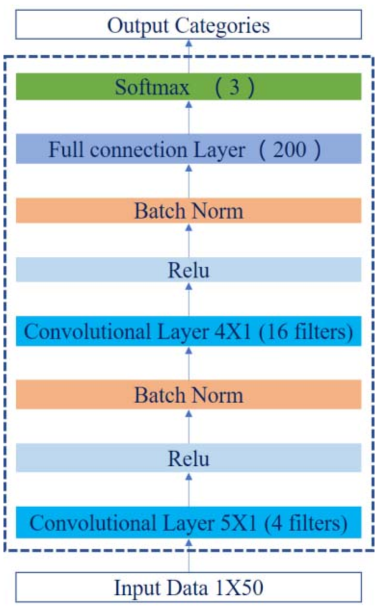


Figure 1. CNN for one dimensional data

Recurrent Neural Network

The concept of the recurrent neural network (RNN) is different from CNN. The hidden layer of RNN is related to time. For the time series data, the input is processed in sequence by different layers. Originally, RNN has the vanishing gradient problem, which means the gradient becomes extremely small with the number of propagation layers. Hence, it is unavoidable if number of the hidden layers are large, which limits the application of RNN. To solve the problem, the *long-short term memory* (LSTM) is proposed, which can be viewed as an enhancement to solve the original RNN problem. The LSTM unit uses memory cells to store information and is described by Figure 3 [17].

The LSTM unit is implemented by following functions: The subscript 'i' denotes the input gate, 'f' the forget gate, 'o' the output gate, and 'c' is the cell.

$$i_t = \sigma\left(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i\right), \tag{6}$$

$$f_t = \sigma\left(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f\right), \tag{7}$$

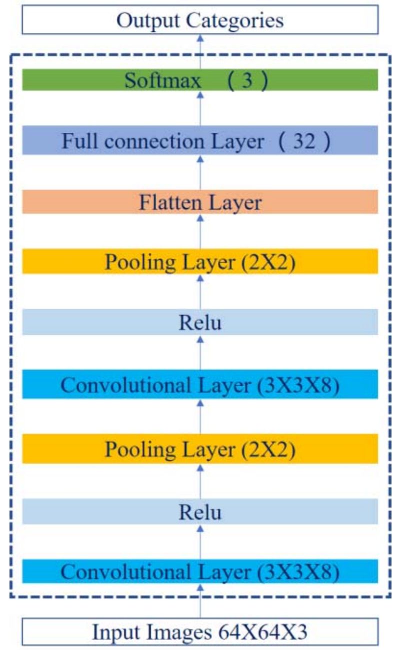


Figure 2. CNN for two-dimensional data

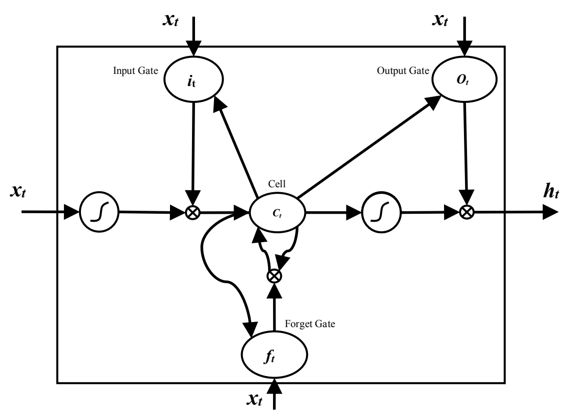


Figure 3. Long Short-term memory cell unit

$$c_t = f_t c_{t-1} + i_t \tanh\left(W_{xc}x_t + W_{hc}h_t + b_c\right), \tag{8}$$

$$o_t = \sigma\left(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o\right), \tag{9}$$

$$h_t = o_t \tanh\left(c_t\right), \tag{10}$$

where $\sigma(\cdot)$ is the logistic sigmoid function, which is given by

$$\sigma(x) = \frac{e^x}{e^x + 1} . \tag{11}$$

W and b denote weight and bias for different terms, respectively, and $\tanh(\cdot)$ is the hyperbolic tangent function.

Note that RNN is very flexible to classification problem, it can be used for 'many to one', 'one to many', 'many to many' classification problem. Further details on CNN, RNN, and LSTM are widely available in the literature.

4. EXPERIMENTS AND TESTING

DATA

Some papers used simulated light curves to test the performance of the classifier. To simulate light curves, a set of parameters are used and the object physical model is built. However, it is difficult to validate the accuracy of the model and it is also hard to obtain parameters for specific space object, such as material and shape. Hence, instead of using synthetic data, real data is used in this paper. The authors realized that it takes long time to collect the light curves and to obtain the phase-folded light curves of resident space objects using public data. Fortunately, the phase-folded star light curves are available. Hence, the star light curves are used instead.

The dataset we used in this paper is from the *Optical Gravitational Lensing Experiment* (OGLE). In the OGLE dataset, millions of stars have been monitored and the light curve data are available [18]. Specifically, the dataset used in [19] is chosen for this study. The dataset includes three different kinds of stars in Magellanic clouds: RRLyrae, eclipsing binaries, and cepheids, The original goal of the dataset is to explore the signal period. However, it is also a good dataset to test classification algorithms [18]. Specifically, the 11142 time series data is used. The ratio of training to test is 7:3. Exemplar light curves for different classes are shown in Figure 4. The data for each column belongs to the same class. The x-axis denotes the observation data index. The y-axis denotes the magnitude.

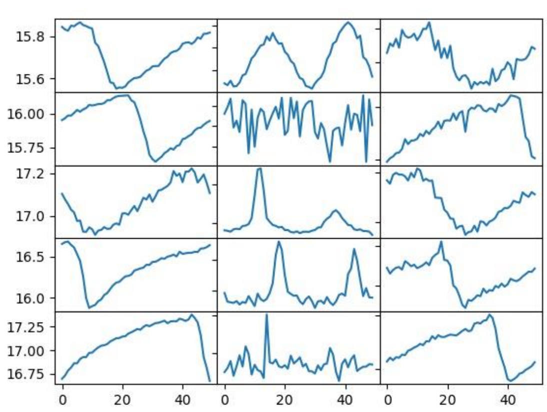


Figure 4. Light curve example

Performance of k-NN-DTW

The algorithm *k*-NN-DTW, shown in Section 2, is used and the accuracy is 0.934 including all categories with k = 1. The performance of *k*-NN-DTW with different *k's* is shown in Figure 5. There is no obvious improvement by using different number of *k*.

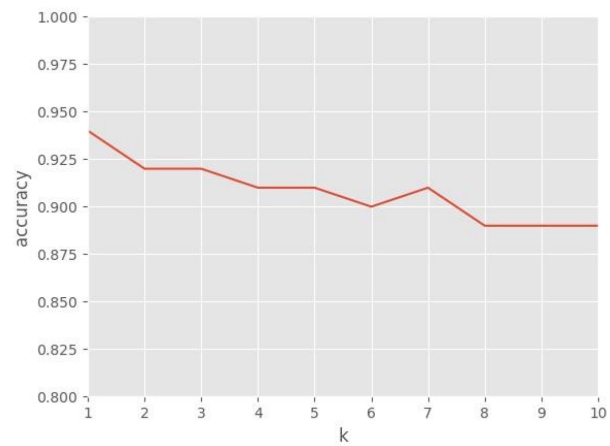


Figure 5. the performance of KNN-DTW with different *k*

Performance of CNN-1D

The algorithm in Section 3.1 is used in this section. Python with Tensorflow is used to implement the neural network. 10,000 training epochs are used. The training and validation accuracy curves are shown in Figure 6. It can be seen that both the training accuracy and the valid accuracy increase with the increasing of the number of training epochs. In addition, the final accuracy on the test data is 0.978, which is significantly better than the conventional k-NN-DTW algorithm of 0.934.

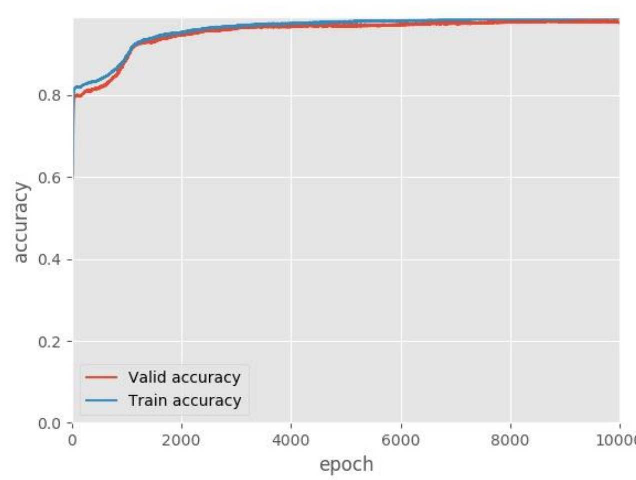
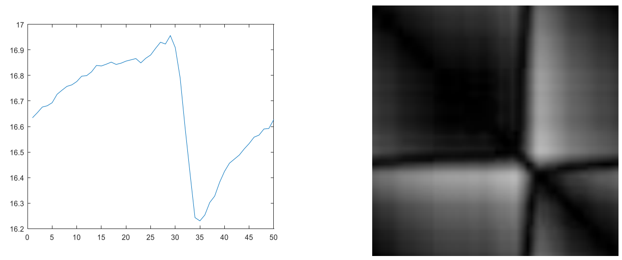


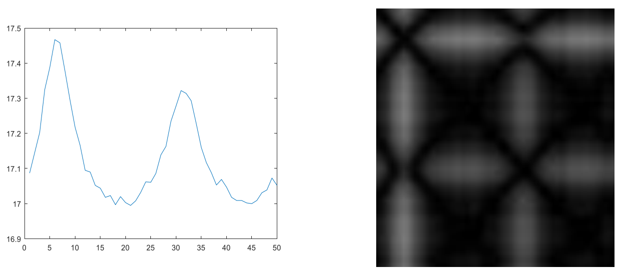
Figure 6. Training and validation accuracy using CNN-1D

Performance of CNN-2D

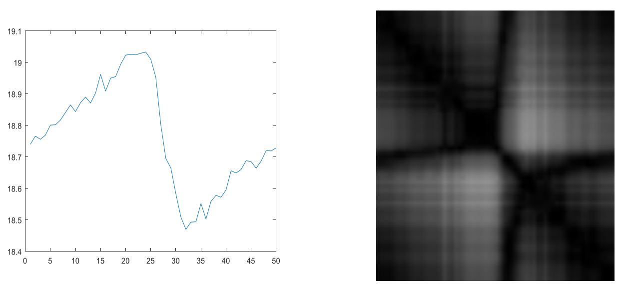
The two-dimensional signal (recurrence plot) is used to in this section. To show how the recurrence plot looks like, three-time series from three different categories are chosen (shown on the left side) and their corresponding recurrence plots are shown in the right side of Figure 7.



(a), Category 1 (b), Category 1



(a), Category 2 (b), Category 2



(a), Category 3 (b), Category 3

Figure 7. Conversion from time series to recurrence plot

It can be seen that for some time series data in category 1 and category 3, they are very close in texture, which is difficult to distinguish.

The training and testing accuracy with the increasing number of training epochs is shown in Figure 8. It can be seen that the accuracy is maintained over 0.96, which is better than k-NN-DTW but inferior than CNN-1D. A possible explanation is that the recurrence plot using Eq. (3) losses the magnitude information of the time series data, which may affect the classification accuracy. Further investigation is needed to incorporate more information in the conversion of the one dimensional signal to a two-dimensional signal.

Performance of RNN

RNN is trained for 500 epochs. To construct RNN, three layers of stacked RNN are used. Each layer of the stacked RNN contains 150 LSTM cells. The learning rate is 0.0005. The batch size is 30. The training and validation accuracy curves are shown in Figure 9. It can be seen that with the increasing number of the training epochs, both the training accuracy and validation accuracy increases. At the final stage, the validation accuracy is in the range from 0.967 to 1, which is higher than that of the conventional *k*-NN-DTW algorithm.

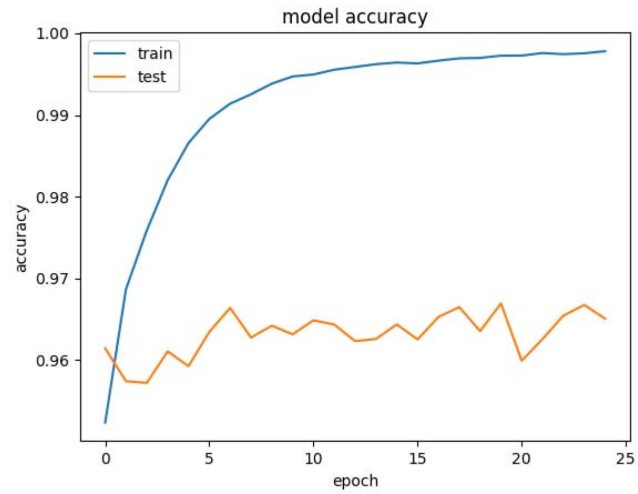


Figure 8. Training and validation accuracy using CNN-2D

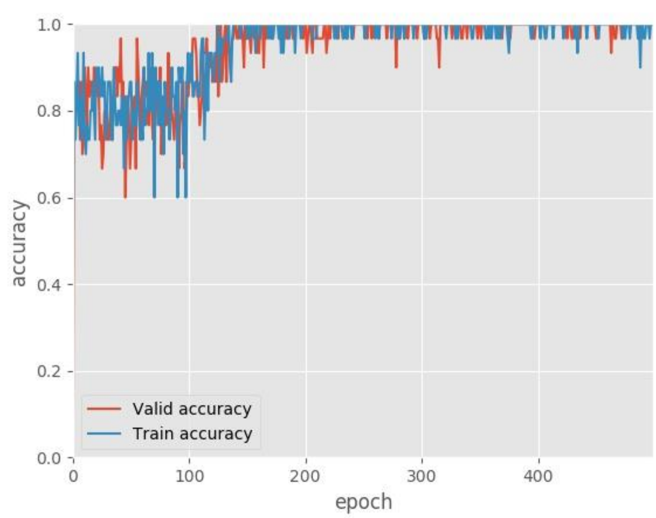


Figure 9. train and valid accuracy using RNN

For convenience, the accuracy of the different classifiers is summarized in Table 1. By comparing the conventional k-NN-DTW with deep neural network algorithms, it can be seen that DNN can lead to more accurate results, which is consistent with expectations. One thing to consider is that training a DNN includes many choices such as the number of layers, and the number of neural units for each layer. When a good parameter set is selected, the performance can improve over conventional methods. The k-NN-DTW algorithm requires only slight tuning of parameters, which is less intensive than DNN. Additionally, the training time of k-NN-DTW is significantly less than that of DNN.

Table 1. Performance of Different Classifiers

Algorithm	Accuracy
k-NN-DTW	0.94
CNN-1D	0.978
CNN-2D	0.962
RNN	0.967

5. SUMMARY

In this paper, we proposed three different NNs to solve the space object classification problem. Both CNN and RNN show promising results as compared to the conventional k-NN-DTW algorithm. In addition, one dimensional time series data is converted to two-dimensional image using the recurrence plot. The two-dimensional CNN also shows promising results, which provides an alternative way to directly use the time series data.

REFERENCES

[1] N. Bobrinsky and L. Del Monte, "The space situational awareness program of the European Space Agency." Cosmic Research, vol. 48, no. 5, 2010, pp. 392-398.

[2] Q. Zhang, H. Wang, R. J. Plemmons, V. P. Pauca, “Tensor methods for Hyperspectral Data Analysis: A Space Object Material Identification Study, “ J. Optical Society of America, A, Vol 25 (12), pp. 3001-3012, 2008.

[3] T. Payne, A. Chaudhary, S. Gregory, J. Brown, M. Nosek, “Signature Intensity Derivative and its Application to Resident Space Object Typing”, Adv. Maui Optical and Space Surveillance Tech. Conf., 2009.

[4] K. J., Astrom, K.J., “On the choice of sampling rates in parametric identification of time series,” Information Sciences, vol.1, no. 3, pp. 273–278, 1969.

[5] E. Keogh and M. Pazzani, “A Simple dimensionality reduction technique for fast similarity search in large time series databases,” In: Proceedings of the 4th Pacific-Asia Conference on Knowledge Discovery and Data Mining, pp. 122–133, 2000.

[6] E. Keogh, “Fast similarity search in the presence of longitudinal scaling in time series databases,” In: Proceedings of the Ninth IEEE International Conference on Tools with Artificial Intelligence, pp. 578–584. 1997.

[7] Chung, F.L., Fu, T.C., Luk, R., Ng, V., “Flexible time series pattern matching based on perceptually important points.” In: International Joint Conference on Artificial Intelligence Workshop on Learning from Temporal and Spatial Data, pp. 1–7, 2001.

[8] R. Agrawal, C. Faloutsos, A. Swami, “Efficient similarity search in sequence databases,” In: Proceedings of the Fourth International Conference on Foundations of Data Organization and Algorithms, pp. 69–84, 1993.

[9] Z. R. Struzik, A.P.J.M. Siebes, “Wavelet transform in similarity paradigm,” In: Proceedings of the Second Pacific-Asia Conference on Knowledge Discovery and Data Mining, 1998, pp. 295–309.

[10] D. J. Berndt, J. Clifford, “Using dynamic time warping to find patterns in time series,” In: AAAI Working Notes of the Knowledge Discovery in Databases Workshop, pp. 359–370, 1994.

[11] K.P., Chan, A. Fu, C. Yu, C., “Haar wavelets for efficient similarity search of time series: with and without time warping.” IEEE Transactions on Knowledge and Data Engineering, vol. 15, no. 3,pp. 685–705, 2003.

[12] F. Tak-chung, A review on time series data mining. pp. 164-181, 2011.

[13] B. Jia, K.D.Pham, E. Blasch, D. Shen, Z. Wang, and G. Chen, "Space object classification using fused features of Time Series Data," 18th Advanced Maui Optical and Space Surveillance Technologies Conference, Maui, HI, 2017.

[14] R. Lindares, J. L., Crassidis, and M. K. Jah, "Space object classification and characterization via multiple model adaptive estimation," 17th International Conference on Information Fusion, 2014, pp. 1-7.

[15] Convolutional neural network, https://en.wikipedia.org/wiki/Convolutional_neural_network

[16] S. Ioffe and C. Szegedy, "Batch normalization: accelerating deep network training by reducing internal covariate shift," ICML'15 Proceedings of the 32nd International Conference on International Conference on Machine Learning, vol. 37, 2015, pp.448-456.

[17] A. Graves, A. Mohamed, G. Hinton, "Speech recognition with deep recurrent neural networks," 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, May, 2013.

[18] S. D. Kügler, N. Gianniotis, and K. L. Polsterer, "Featureless classification of light curves," Monthly Notices of the Royal Astronomical Society, vol. 451, no. 5, pp. 3385-3392, 2015.

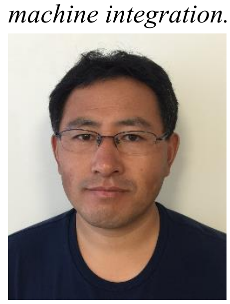
[19] Y. Wang, R. Khardon, and P. Protopapas, "Nonparametric Bayesian Estimation of Periodic Light Curves," The Astrophysical Journal, vol. 756, no.1, 2012.

BIOGRAPHIES

Bin Jia *received his Ph.D. degree in the Department of Aerospace Engineering at Mississippi State University, Starkville, in 2012. He has been with Intelligent Fusion technology, Inc., Germantown, MD as a research scientist from March 2013. His research interests include optimization, statistical signal processing, compressive sensing, nonlinear filtering and estimation, guidance and navigation, and sensor fusion. Prior to IFT, He worked as a Post-Doctoral Research Scientist with the Department of Electrical Engineering, Columbia University, New York, NY.*

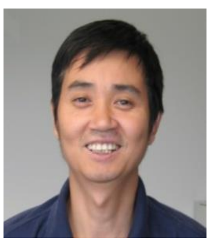
Khanh Pham *serves as a senior aerospace engineer with the Air Force Research Laboratory-Space Vehicles Directorate. He is a senior member of IEEE, SPIE and an associate fellow of AIAA. His BS and MS degrees in Electrical Engineering are from University of Nebraska and PhD in Electrical Engineering from University of Notre Dame. His research interests include statistical optimal control and estimation; fault-tolerant control; dynamic game decision optimization; security of cyber-physical systems; satellite cognitive radios; and control and coordination of large-scale dynamical systems.*

Erik Blasch *received his B.S. in Mechanical Engineering from the Massachusetts Institute of Technology (1992) and PhD in Electrical Engineering from. Wright State University (1999) in addition to numerous Master's Degrees in Mechanical Eng., Industrial Eng., Electrical Eng., medicine, military studies, economics, and business. He has been with the Air Force Research Laboratory since 1996 compiling over 700 papers. He is a Fellow of SPIE, Associate Fellow of AIAA, and a senior member of IEEE. His areas of research include target tracking, image fusion, information fusion performance evaluation, and human-machine integration. space communication systems.*



Zhonghai Wang *received his BS degree from Tianjin University, Tianjin, China in 1998, and Ph.D. degree from Michigan Technological University, MI, USA in 2010. He has been with Intelligent Fusion technology, Inc. (IFT), since March 2012, and* *currently is a senior project manager. Prior to IFT, Dr. Wang worked as a Postdoctoral Research Scientist with the Electrical and Computer Engineering Department at Missouri University of Science and Technology, Rolla, MO, from March 2011 to February 2012. He was an engineer at China Aerospace Science & Industry Corporation from August 1998 to August 2005. His research interests include wireless localization, radar system, array signal processing and statistical signal processing.*

Dan Shen *received the B. S. in Automation from Tsinghua University in 1998, and the M. S. and Ph.D. in Electrical Engineering from the Ohio State University in 2003 and 2006; respectively. He is currently a principal scientist of the Intelligent Fusion Technology, Inc., Germantown MD. His research interest include information fusion, data mining, signal processing, dynamic game theory and applications, cooperative control and decision making, cyber network security, and*



Genshe Chen *received the B. S. and M. S. in electrical engineering, PhD in aerospace engineering, in 1989, 1991 and 1994 respectively, all from Northwestern Polytechnical University, Xian, P. R. China.*

Currently Dr. Chen is the chief technology officer of Intelligent Fusion Technologies, Inc, Germantown, MD. His research interests include cooperative control and optimization for military operations, Target tracking and multi-sensor fusion, Cyber Security, C4ISR, Electronic Warfare, Digital signal processing and image processing, Game theoretic estimation and control, Bayesian network, Influence diagram, robotics, and Human-Cyber-Physical system.