2014 IEEE Congress on Evolutionary Computation (CEC)
July 6-11, 2014, Beijing, China

Vessel Track Correlation and Association using Fuzzy Logic and Echo State Networks

Hang Shao and Nathalie Japkowicz
School of EECS, University of Ottawa
Ottawa, Canada
hshao068@uottawa.ca, nat@eecs.uottawa.ca

Rami Abielmona and Rafael Falcon
Research & Engineering, Larus Technologies Corporation
Ottawa, Canada
rami.abielmona@larus.com, rafael.falcon@larus.com

Abstract— **Tracking moving objects is a task of the utmost importance to the defence community. As this task requires high accuracy, rather than employing a single detector, it has become common to use multiple ones. In such cases, the tracks produced by these detectors need to be correlated (if they belong to the same sensing modality) or associated (if they were produced by different sensing modalities). In this work, we introduce Computational-Intelligence-based methods for correlating and associating various contacts and tracks pertaining to maritime vessels in an area of interest. Fuzzy *k*-Nearest Neighbours will be used to conduct track correlation and Fuzzy C-Means clustering will be applied for association. In that way, the uncertainty of the track correlation and association is handled through fuzzy logic. To better model the state of the moving target, the traditional Kalman Filter will be extended using an Echo State Network. Experimental results on five different types of sensing systems will be discussed to justify the choices made in the development of our approach. In particular, we will demonstrate the judiciousness of using Fuzzy *k*-Nearest Neighbours and Fuzzy C-Means on our tracking system and show how the extension of the traditional Kalman Filter by a recurrent neural network is superior to its extension by other methods.**

Keywords— Computational Intelligence; Track Correlation; Track Association; Data Fusion; Defence and Security; Neural Networks; Fuzzy Logic; Maritime Domain Awareness;

I. Introduction

Tracking moving targets is an important task in defence and security. Such a task can be performed with the help of various kinds of sensors, such as Radar, Global Positioning System (GPS) or Ground Moving Target Indicator (GMTI).

Two tasks involved in the moving target tracking problem are track correlation and association. *Track correlation* is the generation of tracks based on different contact reports from the same sensor that are believed to describe the same object. When there are multiple types of sensors in the tracking system, if various tracks from different sensors are deemed to represent the same object, such tracks can be associated into one single track. The goal of *track association* is to develop tracks based on contact reports obtained from multiple types of sensors. Track correlation is typically done first. Based on the correlation results, track association then follows. The tracks could remain correlated even if an association could not be performed among them.

As traditional and widely used tracking systems, Automatic Radar Plotting Aid (ARPA) radars [1] and the later developed Automatic Identification System (AIS) [2] have been applied in tracking endeavours. Recently, a new tracking format called Ground Moving Target Indicator (GMTI) [3] has been developed. This research will conduct track correlation and association on contact reports from these three types of sensors as well as the Global Positioning System (GPS).

Our work makes the following three contributions:

- We perform track correlation with Fuzzy *k*-Nearest Neighbours (Fuzzy *k*-NN) and apply Fuzzy C-Means (FCM) clustering for association. The advantage of using fuzzy logic for these tasks is that the membership grades to different concepts can help model the certainty of the correlation and association tasks.
- We extend the traditional Kalman Filter (KF) using an Echo State Network (ESN), a particular type of recurrent neural network, to better model the state of the moving target.
- We associate GMTI tracks with ARPA- and AIS-generated tracks. The novelty here lies in the fact that the GMTI technology was never used before within a Computational Intelligence (CI-) based framework, in conjunction with the older ARPA and AIS technologies.

Additionally, we associate Synthetic Aperture Radar (SAR) contacts from Canada's RADARSAT-2 system with AIS tracks. We will demonstrate the judiciousness of using Fuzzy *k*-NN and FCM within our tracking system and show how the extension of the traditional KF by a recurrent neural network is superior to its extension by other methods.

The rest of the article is structured as follows. Section II briefly reviews relevant work whereas Section III introduces the new CI-based track correlation and association methodology. The empirical evaluation is unfolded in Section IV and some concluding remarks are given in Section V.

II. Relevant Works

The existing track correlation/association algorithms can be divided into CI-based and non-CI-based methods.

A. Non-CI-based Track Correlation and Association

The track correlation problem has been studied since the 1970s. Kanyuch and Singer [4] used a simple computer-controlled gate technique where two tracks consecutively falling within these gates, as verified by a number of statistical tests, will be declared as correlated. This simple method, however, was found to degenerate seriously in high-track-density environments. Later on, Bar-Shalom and Tse [5, 6] introduced two types of filters called the Probabilistic Data Association filter (PDA) and the Joint Probabilistic Data Association filter (JPDA) to perform track correlation and association. For these methods, a probability density function has to be estimated to express the uncertainty of the data. Unfortunately, these methods were sensitive to noise and they suffer from large computational complexity as the number of targets increases [7]. It would be computationally intractable when there are too many targets.

The nearest neighbor (NN) algorithm is a well-known, simple and effective non-CI based method that can be used for track correlation/association purposes. The idea is that tracks that are closest to each other will be associated. S. Mori et al. [8] and C-Y. Chong et al. [9] use the Mahalanobis distance as the track association metric. In [8], several other track association metrics and fusion algorithms are compared using the linear Gaussian-Poisson model. However, these comparisons are restricted only to the simplest two-track scenario. It was shown that the NN method, though efficient in low-density cases, is not reliable in high-density scenarios.

Previous works have also produced a number of Maximum- Likelihood-based methods. L. M. Kaplan et al. [10] defined a cost function derived from the Infinite Prior Likelihood (IPL) for multi-sensor track-to-track association. The disadvantage of this method is that in order to calculate the IPL, the prior has to be Gaussian which might not be true in all applications. Another approach by Bar-Shalom and Chen [11] assumed dependency of the estimation error of different sensors on the same target for track-to-track association and derived the track association likelihood from this assumption. From there, a likelihood ratio cost function used in a multidimensional assignment algorithm was proposed.

Furthermore, as a well-known non-CI method, the Kalman Filter (KF) [12, 13] can also be used in the track correlation/association problem. In the KF framework, a state space representation has to be established first. Then, the KF is able to estimate the future state of the target's movement based on current measurement and previous state estimations. Usually, the KF is used to analyze linear Gaussian systems; however, the movement of the target is not linear in most cases. There are many improved versions that were subsequently developed [14], such as the Extended KF and the Unscented KF, which mainly deal with nonlinear cases. In such methods, the nonlinear estimation is approximated by first-order and second-order Taylor expansions or sampling techniques. However, such modifications are much harder to solve. They will inevitably increase the computational complexity of the original KF.

The *M*-out-of-*N* method was also designed for a number of specific sensors and platforms [15, 16]. In the *M*-out-of-*N* method, a cost function is first defined. A pair of tracks that have the minimal cost for *M* samples out of the last *N* associations will be fused into a single track. In this method, the measurements are assumed to be Gaussian and the cost function largely depends on the covariance of the state estimation. However, there is no criterion to decide the optimal value of *M* and *N*. For example, in [15] *M* seems to be arbitrarily set to 3 and *N* is arbitrarily chosen to be 4.

In addition, other non-CI methods such as signal-processing-based techniques [17] and the Distributed Multi-Hypothesis Tracker [18] are also valid in track correlation/association. Such methods have been successfully applied to these types of problems. In [17], the track trajectory is considered to be "a signal in a certain period of time" and it is only able to conduct association. In [18], no evaluation of the fusion is provided.

B. CI-based Track Correlation and Association

A few CI-based approaches that lean on artificial neural networks [19] and fuzzy logic [20] were also used for track correlation and association. The track association problem can be viewed as a multi-dimensional classical assignment problem which could be solved by a Hopfield neural network [21]. Also, the same assignment problem can be solved by a Boltzmann machine [22]. However, such methods depend on an accurate weighted distance matrix which may be hard to get in some cases.

The Multi-Layer Perceptron (MLP) can also be applied to track association [23]. The absolute value of the difference of course, speed, distance and bearing between two vessel tracks can be selected as the input vector. This method is based on the assumption that the smaller the difference, the higher the probability that the tracks belong to the same object. There is also a work combining General Regression Neural Network (GRNN) and the KF [24]. The GRNN is used to perform maneuver detection when the target is changing its speed. The output of the GRNN (i.e., the movement vector) is used in the KF's state updates. The GRNN can only work well on maneuver patterns which are similar to the ones it has seen before. In such cases, the GRNN may need to memorize a large number of patterns, thus slowing down its performance.

The correlation and association performance of classical systems can also be improved using fuzzy logic methods. Previous studies have found that by including properly defined fuzzy sets, Fuzzy Data Association renders better performance than the JPDA filter in certain cases [20]. The FCM clustering algorithm has also been used for the association task [25]. However, the method has to know the number of clusters in advance and is quite sensitive to this parameter. Besides, multi-factor fuzzy integration decision making can also be employed to judge the relevance degree of two tracks [26], but the algorithm is very complex as the fuzzy factor set, judgment matrix and integration evaluation rules have to all be defined. In [27], based on the precision of the data, different association equations are defined for the latitude, longitude, course and speed, and fuzzy association is performed based on the equations. This scheme, though simple and effective, is sensitive to the membership function parameters and thresholds needed to craft the fuzzy rule base.

In this work, the tracking problem is divided into correlation, synchronization, association and filtering. Unfortunately, the existing approaches only solve part of the tracking problems in our system. For example, MLPs [23] can only let you know which tracks should be associated, but it is unable to fuse multiple tracks. *M*-out-of-*N* [15, 16] can only deal with association and GRNN and KF [24] are unable to conduct track correlation. Our contribution consists in filling in the missing parts. In addition, we employ a Recurrent Neural Network (RNN) under the KF framework to model the target movement in presence of nonlinearity. This is a novel algorithm which is shown to be effective and efficient.

III. The New CI-based Method

In this section, a new CI-based correlation and association method is introduced. This method involves several steps including correlation, synchronization, association and filtering. It is worth mentioning that we are not building an entirely new approach. The novelty of this work is a practical and suitable amalgamation of existing methods to solve the correlation and association tasks.

Fuzzy methods will be applied to both track correlation and association in order to gain insight into the certainty of these tasks. The track correlation relies upon the Fuzzy *k*-NN [28] method except for AIS messages, which are correlated via the Maritime Mobile Service Identity (MMSI) field, which is a serial number of nine digits used to uniquely identify a vessel. We are aware that in the real world a vessel's MMSI may be incorrectly transmitted or intentionally spoofed. In such cases, we can use Fuzzy *k*-NN to correlate the AIS track, but in our experiment, we assume that all the MMSIs are reliable as most of the data used are synthetic.

Different sensing systems may have different sampling periods and frequencies. They may also have different coordinate systems. Therefore, data from different sensors have to be synchronized before fusion is attempted. After synchronization, the tracks can be associated by FCM [29]. A joint KF and Recurrent Neural Network (RNN) approach is applied to the associated tracks. As in real-world applications, there will always be a certain amount of noise in the sensor measurements which is why the step of filtering is necessary.

The KF is a well-known method that copes well with linear Gaussian systems. In the real world, however, the movement of the vessel is usually nonlinear. Although there are other KF extensions for nonlinear systems, they inevitably increase the computational complexity of the KF. Moreover, the reason for the nonlinear movement is that there is an additional parameter of the moving target that makes the speed change happen. Usually, such a parameter cannot be directly measured by the sensor, so we need to find a way to model it. RNNs are a good option for dynamic systems. Compared with other KF extensions, we are not making it more complex by introducing nonlinear state transition or measurement matrix. We tackle the problem from its original source of nonlinearity through the learning of the additional input parameter.

The workflow of the proposed CI-based correlation and association method is illustrated in Figure 1. The red block is the filtering component. The rest of this section elaborates on the different building blocks of the proposed scheme.

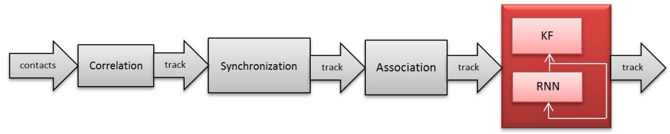

Fig. 1. *The proposed CI-based track correlation/association method*

A. Track Correlation with Fuzzy k-Nearest-Neighbour

In the proposed CI-based method, the correlation, except in the case of AIS messages, is realized through Fuzzy *k*-NN, the fuzzy version of the well-known *k*-Nearest-Neighbour (*k*-NN) algorithm. In *k*-NN, each of the *k* neighbours of the test data sample to classify is assigned equal importance (i.e. weight) in the decision making process whereas in Fuzzy *k*-NN, based on the similarities between the *k* neighbours and test data sample, they may have different weights. Therefore, in Fuzzy *k*-NN, the data points that are more similar to the test sample are then more likely to affect the final classification.

The traditional *k*-NN only makes hard classifications, i.e., no information about the certainty of the classification is provided. The fuzzy version introduces the concept of class memberships which shed light on how confident Fuzzy *k*-NN is about the particular classification.

Given n data samples $\boldsymbol{X}=\{\boldsymbol{x}_1, \boldsymbol{x}_2,\ldots, \boldsymbol{x}_n\}$ and their labels $u_{ij} \in \{0,1\}$, where $u_{ij}=1$ indicates that data vector $\boldsymbol{x}_j$ belongs to class i and $u_{ij}=0$ indicates that it does not belong to class i, the membership of a test vector $\boldsymbol{z}$ being assigned to class i can be calculated by:

$$u_i(\boldsymbol{z}) = \frac{\sum_{j=1}^{k} u_{ij}(\| \boldsymbol{z} - \boldsymbol{x}_j \|^{-2/(m-1)})}{\sum_{j=1}^{k}(\| \boldsymbol{z} - \boldsymbol{x}_j \|^{-2/(m-1)})} \tag{1}$$

where m is the fuzzifier parameter (typically set to 2). The membership grades have to satisfy the following constraint:

$$\sum_{i=1}^{c} u_i(\boldsymbol{z}) = 1 \tag{2}$$

Fuzzy *k*-NN will assign the test data sample to the class with the maximum membership value, as shown below:

Algorithm 1. Fuzzy *k*- Nearest-Neighbour

Input: Labeled Data $\boldsymbol{x}_j$, u_{ij}, i=1,2,..,c, j=1,2,…,n (c is the number of classes and n is the number of samples), k ($0 \le k \le n$), fuzzifier m, test sample $\boldsymbol{z}$
Output: Class membership $u_i(\boldsymbol{z})$
Begin
1: Initialize the k-NN set with the first k data vectors
2: **for** $i = k+1$ **to** n
3:    **if** $\boldsymbol{x}_i$ is closer to $\boldsymbol{z}$ than the farthest of the k-nearest-neighbour
4:       Replace the farthest neighbour with $\boldsymbol{x}_i$
5:    **end if**
6: **end for**
7: **for** i=1 **to** c
8:    Compute $u_i(\boldsymbol{z})$ using (1)
9: **end for**
10: **return argmax**$_i$ $u_i(\boldsymbol{z})$
End

B. Track Association with Fuzzy C-Means Clustering

As previously discussed, the association step will attempt to fuse the correlated tracks, which were obtained through the application of Fuzzy k-NN to different sensing modalities. In our framework, the association is realized by FCM clustering.

1) Synchronization

Note that due to the different coordinate systems and sampling frequencies, it is necessary to conduct space and time synchronization ahead of multi-sensor fusion [27]. The latitude/longitude coordinate system can be transformed to the Cartesian coordinate system via the Universal Transverse Mercator (UTM) projection [30]. The tracks from different sensors will also need to be synchronized in time before association takes place.

2) Fuzzy C-Means Clustering

FCM also deals with the concept of membership when assigning data samples to clusters. That is, the data sample could be assigned to more than one cluster, which is different from hard clustering where each data vector can be associated to only one cluster at a time.

Given n data samples $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots \boldsymbol{x}_n$, the number of clusters c and initial membership values u_{ij}, $0 \le u_{ij} \le 1$, the objective function of FCM clustering can be written as:

$$L(U, \boldsymbol{c}_1, \boldsymbol{c}_2, ..., \boldsymbol{c}_c) = \sum_{i=1}^{c} \sum_{j}^{n} u_{ij}^m d_{ij}^2 \quad s.t. \quad \sum_{i=1}^{c} u_{ij} = 1 \qquad (3)$$

where U is a matrix whose elements are u_{ij}, $\boldsymbol{c}$ is the number of clusters and $d_{ij} = \|\boldsymbol{c}_i - \boldsymbol{x}_j\|$ is the Euclidean distance between the i-th cluster prototype and the j-th data vector.

Through a Lagrange reformulation, we can calculate the cluster prototypes by equation (4):

$$\boldsymbol{c}_i = \frac{\sum_{j=1}^{n} u_{ij}^m \boldsymbol{x}_j}{\sum_{j=1}^{n} u_{ij}^m} \qquad (4)$$

The membership grade u_{ij} can be calculated as in (5):

$$u_{ij} = \frac{1}{\sum_{k=1}^{c} \left(\frac{\| \boldsymbol{c}_i - \boldsymbol{x}_j \|}{\| \boldsymbol{c}_k - \boldsymbol{x}_j \|} \right)^{2/(m-1)}} \qquad (5)$$

The FCM clustering algorithm can be described as follows:

Algorithm 2. Fuzzy C-Means clustering

Input: Data samples $\boldsymbol{x}_j$, u_{ij}, i=1,2,..,c, j=1,2,…,n (c is the number of classes and n is the number of samples), number of clusters c, matrix U, fuzzifier m
Output: Clustering centers $\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_c$ and membership U
Begin
1: Set $i = 1$
2: **while** $\Delta L(U, \boldsymbol{c}_1, \ldots, \boldsymbol{c}_c) < \varepsilon$ and $i <$ max number of iterations
3: calculate $\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_c$ using (4)
4: calculate u_{ij} using (5)
5: calculate $L(U, \boldsymbol{c}_1, \ldots, \boldsymbol{c}_c)$ using (3)
6: Set $i = i + 1$
7: **end while**
End

In the track association algorithm, the number of cluster prototypes c is set to be the number of unique MMSIs within the AIS tracks. In a real maritime application, if there is a **dark target**, i.e. a vessel that does not advertise itself on AIS but is detected by another sensor, for example radar, the latter track will not be associated with the existing AIS tracks. It will remain as a correlated radar track. In other cases, wherever we are provided with no AIS tracks (for instance, we are associating only radar tracks and GPS tracks), the value c can be set to either the number of radar tracks or GPS tracks. To make full use of correlated track features $\boldsymbol{f}$, the initial u_{ij} is calculated by:

$$u_{ij} = \frac{\| \boldsymbol{f}_i - \boldsymbol{f}_j \|^{-1}}{\sum_{k=1}^{c} \| \boldsymbol{f}_k - \boldsymbol{f}_j \|^{-1}} \qquad (6)$$

The feature set $\boldsymbol{f}$ used in this work represents a 9-dimensional vector that includes: track starting position, ending position, duration, maximum latitudinal/longitudinal position, minimum latitudinal/longitudinal position, standard deviation of latitudinal/longitudinal position, maximum velocity, minimum velocity and standard deviation of the velocity.

C. Track State Estimation with a Kalman Filter

KF is a quite popular method to model linear Gaussian systems, i.e. those in which the state transition and measurement equations are all linear and the noise processes are assumed to be Gaussian. When modeling the movement of a vessel with KFs, its location and speed are usually set as the state space variables. The discrete state transition model can be written as follows [15, 24]:

$$\boldsymbol{x}_k = \boldsymbol{F}\boldsymbol{x}_{k-1} + \boldsymbol{w}_k \qquad (7)$$

where $\boldsymbol{x}$ is the state vector and $\boldsymbol{w}_k \sim N(0, \boldsymbol{Q}_k)$ is the noise process. The transition matrix $\boldsymbol{F}$ can be decided by the physical laws of motion. Typically, speed is considered to be constant within one sample period (interval between two measurements).

When the movement of the target is uniformly rectilinear, the transition model in (7) is suitable. However, resulting from additional control inputs, the vessel movement is nonlinear during maneuvers when the speed is changing. The speed estimation based on the above model will be inaccurate, hence leading to further errors in the target position estimation. It is necessary to include the additional input $\boldsymbol{u}$ into (7),

$$\boldsymbol{x}_k = \boldsymbol{F}\boldsymbol{x}_{k-1} + \boldsymbol{B}\boldsymbol{u}_k + \boldsymbol{w}_k \qquad (8)$$

where $\boldsymbol{B}$ is the control input matrix. The observation model can now be expressed as:

$$\boldsymbol{z}_k = \boldsymbol{H}\boldsymbol{x}_{k-1} + \boldsymbol{v}_k, \ \boldsymbol{v}_k \sim N(\boldsymbol{0}, \boldsymbol{R}_k) \qquad (9)$$

where $\boldsymbol{H}$ is the observation matrix, $\boldsymbol{z}_k$ is the measurement and $\boldsymbol{v}_k$ is the measurement noise.

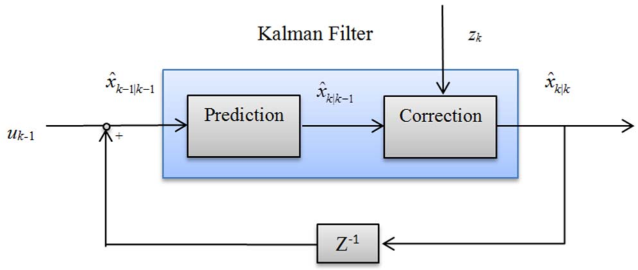

Fig. 2. *The traditional Kalman Filter*

Figure 2 shows the KF and the system defined by (8). The Z^{-1} is the discrete time delay operator. The formulators involved in the Kalman Filter can be listed as follows [27]:

$$\hat{\boldsymbol{x}}_{k|k-1} = \boldsymbol{F}\hat{\boldsymbol{x}}_{k-1|k-1} + \boldsymbol{B}\boldsymbol{u}_{k\text{-}1} \tag{10}$$

$$\boldsymbol{P}_{k|k-1} = \boldsymbol{F}\boldsymbol{P}_{k-1|k-1}\boldsymbol{F}^{\mathrm{T}} + \boldsymbol{Q}_k \tag{11}$$

$$\hat{\boldsymbol{y}}_k = \boldsymbol{z}_k - \boldsymbol{H}\hat{\boldsymbol{x}}_{k|k-1} \tag{12}$$

$$\boldsymbol{S}_k = \boldsymbol{H}\boldsymbol{P}_{k|k-1}\boldsymbol{H}^{\mathrm{T}} + \boldsymbol{R}_k \tag{13}$$

$$\boldsymbol{K}_k = \boldsymbol{P}_{k|k-1}\boldsymbol{H}_k^{\mathrm{T}}\boldsymbol{S}_k^{-1} \tag{14}$$

$$\hat{\boldsymbol{x}}_{k|k} = \hat{\boldsymbol{x}}_{k|k-1} + \boldsymbol{K}_k\hat{\boldsymbol{y}}_k \tag{15}$$

$$\boldsymbol{P}_{k|k} = (\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{H}_k)\boldsymbol{P}_{k|k-1} \tag{16}$$

where $\boldsymbol{P}$ is the covariance matrix of the state estimation error and $\boldsymbol{P}_{0|0}=\mathrm{cov}(\boldsymbol{x}_0\text{-}\boldsymbol{x}_{0|0})$. $\boldsymbol{y}$ is the residual, $\boldsymbol{S}$ is the covariance of residual, $\boldsymbol{K}$ is the Kalman gain and $\boldsymbol{I}$ is the identity matrix.

Equations (10) and (11) model the prediction step in Figure 2 and (12) - (16) describe the correction step where the new measurement z_k is used to update the state estimation. It is noticed from (9) that the control input $\boldsymbol{u}$ cannot be measured by $\boldsymbol{z}$, therefore, we need to find a method to estimate its value. This is where another CI-based method using a recurrent neural network (RNN) comes into play.

D. Echo State Network

In a static network such as MLP or GRNN, the network input could be unrelated to the previous input/state of the network. To model a dynamic system, RNNs are a better choice than static networks. An Echo State Network (ESN) is a type of RNN [32, 33] whose architecture is given in Figure 3. Notice that the ESN topology contains feedback connections.

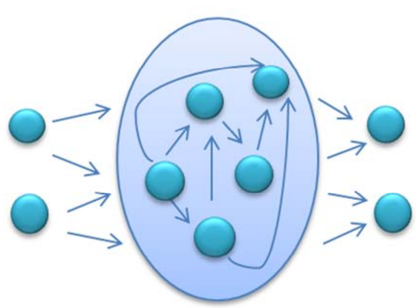

Fig. 3. *Echo State Network*

Like other networks, the only visible neurons in an ESN are the input and output neurons. The internal neurons form a large dynamic reservoir. The neurons inside the reservoir are sparsely connected. Given the input vector $\boldsymbol{in}$ and the corresponding label $\boldsymbol{t}_i$, the state update of ESN can be expressed as

$$\begin{aligned}\boldsymbol{x}_{k+1} &= f(\boldsymbol{in}_k, \boldsymbol{x}_k, \boldsymbol{W}_x, \boldsymbol{W}_{in}) \\ &= \mathbf{tansig}(\boldsymbol{W}_x \cdot \boldsymbol{x}_k + \boldsymbol{W}_{in} \cdot \boldsymbol{in}_k)\end{aligned} \tag{17}$$

where $\boldsymbol{x}_k$ is the state vector of the reservoir, $\boldsymbol{W}_{in}$ is the input weights and $\boldsymbol{W}_x$ are the inner connection weights. Since the reservoir in an ESN is random, the $\boldsymbol{W}_{in}$ and $\boldsymbol{W}_x$ are randomly assigned and do not need training. To ensure stability, the spectral radius of $\boldsymbol{W}_x$ is usually set to a number slightly less than 1. The neuron activation function is defined as:

$$\mathrm{tansig}(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{18}$$

The output of ESN can thus be written as:

$$\boldsymbol{y}_k = \boldsymbol{W}_{out} \cdot \boldsymbol{x}_k \tag{19}$$

where $\boldsymbol{W}_{out}$ is the readout weights. After feeding all of the n input data samples into the ESN, the state of the network $\boldsymbol{x}_1$, $\boldsymbol{x}_2$,,…, $\boldsymbol{x}_n$ can be stored in the state matrix $\boldsymbol{X}=[\boldsymbol{x}_1, \boldsymbol{x}_2,\ldots, \boldsymbol{x}_n]^{\mathrm{T}}$. The output weights $\boldsymbol{W}_{out}$ can be calculated by

$$\boldsymbol{W}_{out} = (\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X} + \lambda\boldsymbol{I})^{-1}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{T} \tag{20}$$

where $\boldsymbol{T}=[\boldsymbol{t}_1,\boldsymbol{t}_2,\ldots,\boldsymbol{t}_n]^{\mathrm{T}}$ are the labels. In the above equation, the parameter λ is added to prevent the matrix inverse from being ill-conditioned.

The ESN learning steps can be summarized as follows:

Algorithm 3. Echo State Network
Input: Data vectors ($\boldsymbol{in}_i$, $\boldsymbol{t}_i$), i=1,2,..,n, L reservoir neurons
Output: Echo State Network
Begin
1: initialize weight matrix $\boldsymbol{W}_{in}$
2: initialize the interconnection matrix $\boldsymbol{W}_x$
3: initialize the state matrix $\boldsymbol{X}$
4: **for** i = 1 **to** n
5: update the state of the reservoir using (17)
6: update the state matrix $\boldsymbol{X}$
7: **end for**
8: calculate $\boldsymbol{W}_{out}$ using (20)
End

The role of the ESN in our approach is to estimate the additional input parameter u_k in the KF. The input vector of the ESN is (*Δposition*, *Δvelocity*) and $\boldsymbol{u}_{k\text{-}1}$ (here $\boldsymbol{u}$ refers to the variable in section III-C) and the output is the estimated $\boldsymbol{u}_k$.

As previously mentioned, there exist nonlinear KF versions such as the Unscented KF and Extended KF. Such methods achieve nonlinearity using Taylor expansion or sampling techniques. Consequently, more computational complexity is introduced to the KF. The source of vessel maneuver is the additional control input $\boldsymbol{u}$ which cannot be usually measured by the sensors. In our approach, the ESN is used to directly model and predict this control input. Neither Unscented KF nor Extended KF has the ability to predict this parameter.

Notice that after filtering the final correlated/associated tracks, it is necessary to apply the reverse UTM projection to transform the tracks back to the latitude/longitude coordinate system. The steps involved in the proposed CI-based method can be summarized as follows:

1. Use Fuzzy *k*-NN to conduct correlation on Radar/GPS/GMTI contacts. Conduct AIS correlation based on a vessel's MMSI;
2. Use UTM projection to convert latitude/longitude coordinates to Cartesian coordinates and perform synchronization;
3. Apply FCM Clustering to associate the AIS/GPS/Radar/GMTI tracks
4. Use ESN to predict the input parameter for the KF;
5. Execute the KF with the output of the ESN to estimate the true state of target movement on associated tracks.
6. Apply reverse UTM projection to convert from Cartesian coordinates back to the latitude/longitude coordinates.

Note that steps 4 and 5 are also valid for correlated tracks. In the real world, there are cases where contacts a certain sensor are very sparse that no correlated tracks can be produced. Such contacts can be directly associated with existing tracks, which is the case in our second experiment.

IV. Experimental Results

The new CI-based correlation/association method was tested on synthetic AIS, ARPA Radar, GPS, SAR and GMTI data. The Area of Interest (AOI) is shown in Figure 4. There are 14 vessels in total in the simulation scenario. For AIS/Radar/GPS, the start and end times were 170000 and 215851, respectively (i.e. 5:00.00 PM and 9:58.51 PM) on January 24th, 2013.

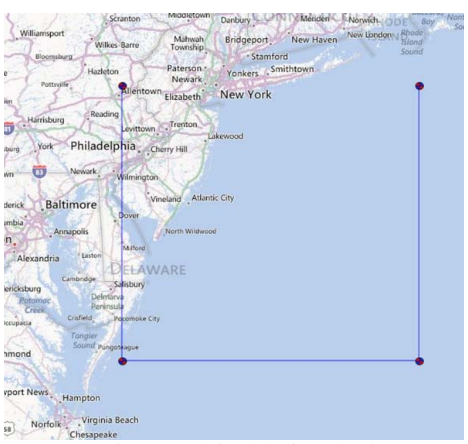

Fig. 4. *Area of Interest defined by four (lat, lon) corners: (40.723469,-75.435744), (40.723469, -70.982369), (37.525305, -70.982369), (37.525305, -75435744)*

As previously mentioned, the correlation for AIS is conducted via matching of the vessel MMSIs. The GPS, Radar and GMTI correlation is performed with the proposed CI-based method. The correlation results are shown in Figures 5 to 8. The correlation method is able to correctly form tracks from discrete contacts from the contacts of various sensors. The correlated tracks clearly show the path of each vessel. Note that for the GMTI tracks, the air platforms used to perform the detections could only see a small part of the track, so the correlated GMTI tracks are very short. The GMTI tracks are first associated to AIS tracks and then the AIS, Radar and GPS tracks are brought in. The final associated tracks are shown in Figure 9 after applying KF and ESN. It is found that the association algorithm is able to fuse multiple tracks from different sensors representing the same vessel into a single associated track. Multiple tracks are shown in different colours in Figures 5 to 9. Each track is made up of many contacts represented by discrete points in the figures.

We also calculate the average value of the maximum membership grades in Fuzzy *k*-NN when the Radar or GPS contact reports correlated to any of the tracks. The average maximum membership grade for both Radar and GPS correlation is 0.9988. In association, the average value of the maximum membership in FCM clustering is 0.9999. Therefore, the two fuzzy methods demonstrate a high certainty. It is worth pointing out that non-fuzzy correlation and association methods are unable to provide such information.

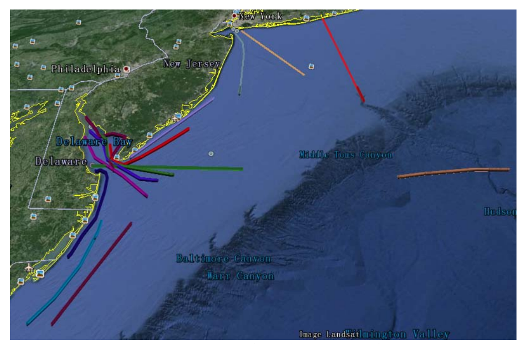

Fig. 5. *Correlated AIS Tracks*

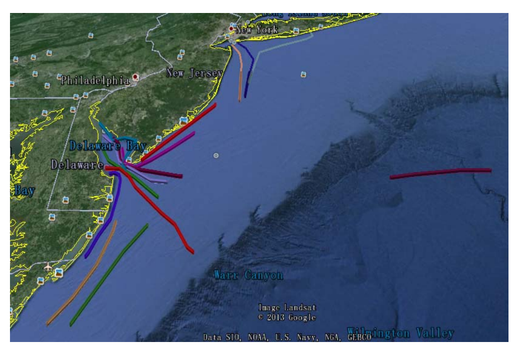

Fig. 6. *Correlated Radar Tracks*

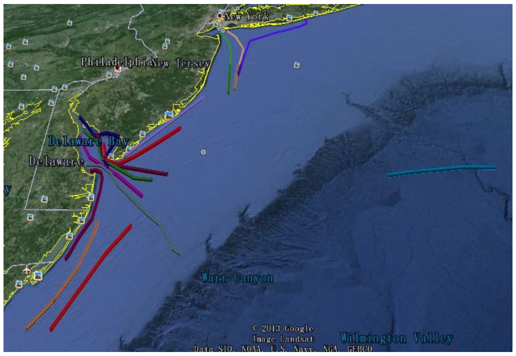

Fig. 7. *Correlated GPS Tracks*

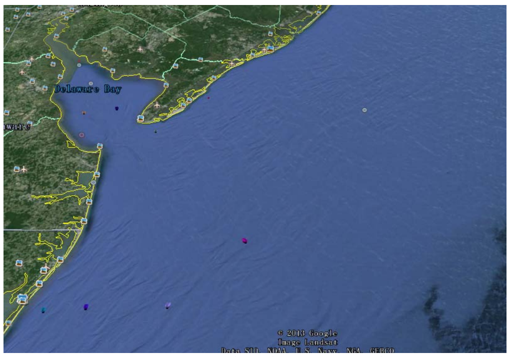

Fig. 8. *Correlated GMTI Tracks*

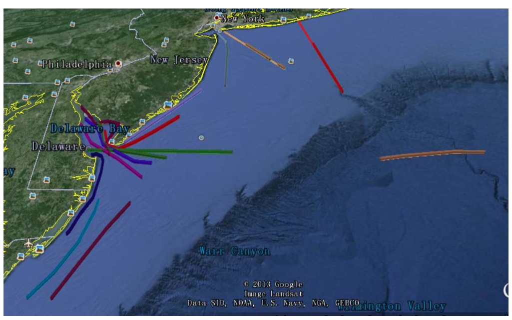

Fig. 9. *Associated Tracks*

TABLE I. PERFORMANCE COMPARISON OF ESN, GRNN AND NARX ON KF-BASED CONTROL INPUT ESTIMATION

Model	**RMSE (std.)**	**Total Time(s)**
ESN	0.1649 (0.00759)	53.87
NARX	0.4180 (0.2390)	31.49
GRNN	0.1410 (0.0755)	4306.19

We also compared the performance of ESN on the KF control input estimation to those of two other neural network models, namely General Regression Neural Network (GRNN) [34] and Nonlinear Auto-Regressive eXogenous model (NARX) [35]. GRNN is a four-layer feedforward network that does not have a free parameter to define the architecture of GRNN. The first layer is the input layer and pattern neurons in the second layer store the input patterns, so the number of the nodes in the second layer is equal to the number of training instances. The third layer has two summation neurons that compute the numerator and denominator for the final output layer. The NARX is type of recurrent network where there are feedback connections between the output layer and the input layer. The number of hidden nodes is optimized from [5, 10, 15,…, 100], and the best results are reported in Table 1 (where 5 hidden nodes are used). ESN is not sensitive to the network size. The number of nodes in the reservoir is set to 100 without optimization. All the networks are trained on the first 20 minutes of the associated tracks and tested on the remaining part of the tracks. The results are shown in Table I. The Root Mean Square Error (RMSE) and its standard deviation are reported. The time recorded in the table includes both training and testing time. From Table 1, we could see that the NARX displays the worst performance (largest RMSE) and the large standard deviation indicates that its performance is unstable. Since the network size of NARX is small, it is the fastest. GRNN has the lowest RMSE. Unfortunately, it stores all the input data in the second layer, making it every slow to run, especially on large dataset. ESN has a performance close to GRNN while being much faster than GRNN. ESN is the best choice in the real application where we often deal with large datasets.

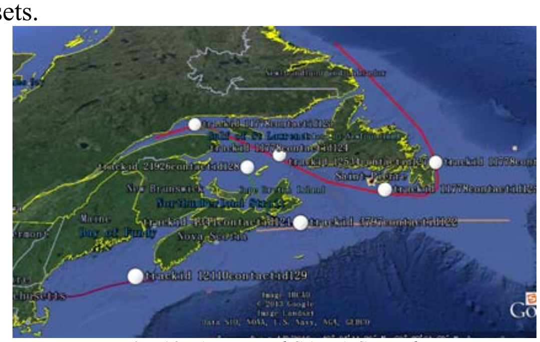

Fig. 10. *Associated SAR-AIS Tracks*

In addition, the new CI-based method was also tested on AIS and Synthetic Aperture Radar (SAR) track association. As there were few SAR contact reports, no correlated tracks were developed; instead, the SAR contacts were directly associated to AIS tracks using Fuzzy *k*-NN. The results are portrayed in Figure 10, with the white circles denoting the SAR-contact-to-AIS-track associations. All SAR contacts were correctly associated with their respective AIS tracks. Since this involved real-world data, dark targets (i.e. SAR contacts that could not be associated with an AIS track) could not be found within the available datasets.

V. CONCLUSION AND FUTURE WORK

This work proposed a new CI-based method for track correlation and association. For AIS contacts, the correlation was done via MMSI, while for other contacts the correlation was performed by Fuzzy *k*-NN. After synchronization, FCM clustering was applied for track association. Finally, the KF was aided by an ESN to help model the nonlinear movement of maritime vessels. ESN turned out to be more efficient than GRNN and more accurate than NARX given that RNNs are more suitable to model dynamic systems. Experimental results show that the new CI-based method is valid and effective in correlating and associating AIS, Radar, GPS, SAR and GMTI tracks. The two fuzzy logic methods, viz. Fuzzy *k*-NN and FCM clustering, demonstrated a high certainty in track correlation and association. Additionally, the ESN is able to provide fast and effective parameter estimation under the KF framework. When the number of vessels increases and the size of the data scales up, the correlation complexity can be manually controlled by limiting the value of k in Fuzzy k-NN. Regarding association and filtering, they will grow linearly with the number of tracks.

As a future work, we will improve Fuzzy k-NN with an adaptive scheme to calculate the optimal number of potential tracks a new contact may belong to. Another future direction could be track normalcy modeling. In this way, vessels with anomalous behaviour could be detected and reported to the corresponding maritime authorities.

REFERENCES

[1] Automatic Radar Plotting Aid, Reference available at http://en.wikipedia.org/wiki/Automatic_radar_plotting_aid".

[2] B.J. Tetreault, "Use of the Automatic Identification System (AIS) for maritime domain awareness (MDA)", OCEANS, Proceedings of MTS/IEEE, 2005.

[3] M Hernandez, "Performance bounds for GMTI tracking", Information Fusion, Proceedings of the Sixth International Conference of, pp. 406-413, 2003.

[4] A.J. Kanyuch, R.A. Singer, "Correlation of Multiple-Site Track Data", Aerospace and Electronic Systems, IEEE Transactions on , pp. 180-187, 1970.

[5] Y. Bar-Shalom and E. Tse, "Tracking in a Cluttered Environment with Probabilistic Data Association," Automatica, vol. 11, pp. 451-460, 1975.

[6] Y. Bar-Shalom, "Extension of the Probabilistic Data Association Filter to Multitarget Tracking," Proc. Fifth Symp. Nonlinear Estimation, pp. 16-21, Sept. 1974.

[7] R.J. Fitzgerald, "Development of Practical PDA Logic for Multitarget Tracking by Microprocessor," Multitarget Multisensor Tracking: Advanced Applications, Y. Bar-Shalom, ed., 1989.

[8] S Mori, W.H. Barker, C-Y. Chong, K-C. Chang, "Track Association and Track Fusion with Non-Deterministic Target Dynamics", Aerospace and Electronic Systems, IEEE Transactions on , vol. 38, issue. 2, pp. 659 -668, 2002.

[9] C-Y Chong, S Mori, W.H. Barker, K-C. Chang, "Architectures and Algorithms for Track Association and Fusion", Aerospace and Electronic Systems Magazine, IEEE , vol. 15, issue. 1, pp. 5 -13, 2000.

[10] L. Kaplan, Y. Bar-Shalom, W. Blair, "Assignment Costs for Multiple Sensor Track-to-Track Association", Aerospace and Electronic Systems, IEEE Transactions on , vol. 44, issue. 22, pp, 655 – 677, 2008.

[11] Y. Bar-Shalom, H. Chen, "Multisensor Track-to-Track Association for Tracks with Dependent Errors", Decision and Control, 2004. CDC. 43rd IEEE Conference on, vol.3, pp. 2674 – 2679, 2004.

[12] R. Kalman, "A New Approach to Linear Filtering and Prediction Problems," Trans. ASME, J. Basic Eng., no. 82, pp. 34-45, 1960.

[13] J.B. Gao and C.J. Harris, "Some remarks on Kalman filters for the multisensor fusion", Information Fusion, vol. 3, pp. 191-201, 2002.

[14] S. J. Julier and J. K. Uhlmann " Unscented filtering and nonlinear estimation", Proceedings of the IEEE, vol. 92, no. 3, pp.401-422, 2004.

[15] D. Danu, A. Sinha, T. Kirubarajan and M. Farooq, "Fusion of Over-the-Horizon Radar and Automatic Identification Systems for Overall Maritime Picture", Information Fusion, 2007 10th International Conference on, pp. 1-8, 2007.

[16] Radar Corresponding Group, RCG-45 ImpulseNoise Interference Effect on Binary Detection, Contributions 2006, http://www.its.bldrdoc.gov/meetings/rcg/contributions_2006/contributions_2006.php.

[17] J. Shi, B. Hao, F. Zhang, "Track Correlation Based on A'trous algorithm", Fuzzy Systems and Knowledge Discovery, 2009. FSKD '09. Sixth International conference on, vol. 7, pp. 392-396, 2009.

[18] C. Carthel, S. Coraluppi and P. Grignan "Multisensor Tracking and Fusion for Maritime Surveillance," Information Fusion, 2007 10th International conference on pp.1-6, 2007.

[19] M. Winter and G. Favier, "A Neural Network for Data Association," Proc. 1999 IEEE Int'l Conf. Acoustics, Speech, and Signal Processing, vol. 2, pp. 1041-1044, Mar. 1999.

[20] H. Hong, H. Chong-zhao, Z. Hong-Yan, and W. Rong, "Multi-Target Tracking Based on Multi-Sensor Information Fusion with Fuzzy Inference," Control and Decision, vol. 19, no. 3, pp. 272-276, 2004.

[21] S. Yang, "ANN-based Track Correlation Algorithm in Multisensor System", The Eighth International Conference on Electronic Measurement and Instruments, pp. 154-158, 2007.

[22] J. Zhang, J. Chen and M. Duan, "Track Correlation for Multisensor System based on NN and General Classical Assignment Algorithm", Proceedings of the 27th Chinese Control Conference, July 16-18, 2008, Kunming,Yunnan, China, pp. 803-806, 2008.

[23] X. Hu and C. Lin, "A Preliminary Study on Targets Association Algorithm of Radar and AIS Using BP Neural Network", Procedia Engineering, vol. 15, pp. 1441-1445, 2011.

[24] A. Stateczny and W. Kazimierski, "Multisensor Tracking of Marine Targets –Decentralized Fusion of Kalman and Neural Filters", International Journal of Electronics and Telecommunications, vol.57, pp. 65–70, 2011.

[25] J. Suo and X. Liu, "Fusion of radar and AIS data", Signal Processing, 2004. Proceedings. ICSP '04. 2004 7th International Conference on, vol. 4, pp. 2604-2607, 2004.

[26] C. Lin, F. Dong, H. Lin, L. Le, J. Zhou and Y. Ou, "AIS Information Decoding and Fuzzy Fusion Processing with Marine Radar", Wireless Communications, Networking and Mobile Computing, 2008. WiCOM '08. 4th International Conference on, pp. 1-5, 2008.

[27] C. Liu and X. Shi, "Study of Data Fusion of AIS and Radar", International Conference of Soft Computing and Pattern Recognition, pp. 674-677, 2007.

[28] J.M. Keller, M.R. Gray and J.A. Givens, "A fuzzy K-nearest neighbor algorithm", Systems, Man and Cybernetics, IEEE Transactions on, vol. 15, issue 4, pp.580-585, 1985.

[29] J. C. Bezdek, "Pattern Recognition with Fuzzy Objective Function Algorithms", Plenum Press, New York, 1981.

[30] R. L. Edwards, "Archaeological Use of the Universal Transverse Mercator Grid", American Antiquity Vol. 34, No. 2, pp. 180-182, 1969.

[31] R.E. Kalman, "A New Approach to Linear Filtering and Prediction Problems", Transactions of the ASME - Journal of Basic Engineering Vol. 82, pp. 35-45, 1960.

[32] H. Jaeger and H. Haas, "Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication", Science, 304, pp.78-80, 2004.

[33] H. Jaeger, "The "echo state" approach to analysing and training recurrent neural networks-with an erratum note," Technical Report GMD Report 148, German National Research Center for Information Technology, 2001.

[34] D. F. Specht, "A general regression neural network", Neural Networks, IEEE Transactions on, vol 2 , issue 6, pp. 568 – 576, 2002.

[35] T. Lin, B. G. Horne, P. Tino, and C. L. Giles, "Learning long-term dependencies in NARX recurrent neural networks", IEEE Transactions on Neural Networks, vol. 7, no. 6, pp. 1329-38, Jan. 1996.