

Received August 25, 2020, accepted September 9, 2020, date of current version October 12, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3025144

# A CNN-LSTM Network for Augmenting Target Detection in Real Maritime Wide Area Surveillance Radar Data

ZACHARY BAIRD<sup>1,2</sup>, (Student Member, IEEE), MICHAEL K. MCDONALD<sup>1</sup>,  
SREERAMAN RAJAN<sup>2</sup>, (Senior Member, IEEE), AND SIMON J. LEE<sup>3</sup>

<sup>1</sup>Defence Research and Development Canada (DRDC), Ottawa, ON K1N 1J8, Canada

<sup>2</sup>Carleton University, Ottawa, ON K1S 5B6, Canada

<sup>3</sup>Denbridge Marine Ltd., Birkenhead CH41 5LH, U.K.

Corresponding author: Zachary Baird (zach.baird@cmail.carleton.ca)

This work was supported in part by the Defence Research Development Canada, NSERC, and in part by the North Atlantic Treaty Organization (NATO).

**ABSTRACT** Typical radar detectors exploit only a small proportion of the valuable information contained in radar reflections, i.e. magnitude and Doppler. A neural network-based approach for augmenting traditional radar detector structures using machine learning (ML) is proposed in this paper. Specifically, the network is designed to augment target detection in the field of maritime wide area surveillance for non-coherent data. A combination network consisting of a convolutional neural network (CNN) to extract spatial features and a long short-term memory (LSTM) for extracting temporal patterns in the spatial features is proposed. The network augments the detector structure by blanking out regions of the frame which are classified as not containing a target, thus reducing false alarms. The network is tested on data containing four marine targets collected by a ground-based radar. The data set was chosen because it contains strong sea clutter returns. When ML is used, the receiver operating characteristic (ROC) curves are shifted to lower probability of false alarm (PFA). A Kalman filter tracker was applied to the ML-augmented and baseline detections, and it was shown that ML-augmented detections produced similar tracks at lower PFA. The feature discovering capability of the network is analyzed through a series of tests, and the argument is made that the CNN-LSTM network presented in this work demonstrates the ability to improve the detection performance by exploiting spatial and temporal information in the data.

**INDEX TERMS** CNN, LSTM, maritime surveillance, wide area surveillance radar.

## I. INTRODUCTION

The purpose of this work is to study the use of machine learning (ML) as a means to augment traditional detector structures in situations where targets are not easily separable from surrounding sea clutter due to low signal to clutter plus noise ratio (SCNR) resulting from unfavorable geometries (i.e. high grazing angles) or low observable targets (including stealth vessels or semi-submersibles) [1], [2]. Two aims of this work are to exploit spatial information (aim 1) and temporal information (aim 2) which are neglected by traditional detectors. In particular, two types of ML networks from the field of deep learning, convolutional neural networks (CNN) and long

short-term memory (LSTM), will be examined because of their inherent ability to discover spatial and temporal patterns.

Traditionally, automated target detectors use only amplitude and/or Doppler information [3]. Radar data contains a lot of information which is not being exploited by basic detectors, including spatial patterns (shapes of targets, land clutter, wave fronts) and temporal patterns (persistence and movement of targets and wave fronts, and lack of movement of land clutter). It is these spatial and temporal patterns which are being recognized by human operators for manual detection of targets in radar data. Furthermore, spatial and temporal patterns are more robust than simple amplitude statistics in environments with low SCNR. In some environmental conditions (such as rough sea states or high grazing angles), automatic target detection fails to detect targets, or detection is only possible at unacceptably high false alarm rates.

The associate editor coordinating the review of this manuscript and approving it for publication was Sudipta Roy<sup>1</sup>.

ML is proposed as a means to augment traditional detectors with information obtained by exploiting spatial and temporal patterns within the data. Neural networks can be designed to exploit two-dimensional and temporal patterns, thus using the same information as human operators. ML is investigated as a way to bridge the performance gap between manual detection via human operators and automated detection via traditional detectors.

#### A. USE OF ML FOR RADAR SURVEILLANCE IN LITERATURE

The use of ML or artificial intelligence to aid in radar detection of targets embedded in clutter has been discussed in literature since at least the early 1990s. In [4] artificial intelligence was presented as a means to boost the robustness of the constant false alarm rate (CFAR) algorithm with wide area surveillance (WAS) radar. Clutter was observed in real time and prior knowledge of semantic labeling of clutter was used to choose correct clutter statistics. In [5] and [6] ML-based detection strategies for passive radar were presented. In [7] the micro-Doppler characteristics of a target were used for classification of target types (background clutter, real human target and single simulated tone) using a X-band wide area airborne radar. The target micro-Doppler signature was extracted from a time-frequency plot of received data and a simple AM-FM model was used to fit to the data. The parameters of the model were used as features for training and validating a support vector machine (SVM) classifier. This work used short coherent processing interval (CPI) lengths, which is a limitation of WAS. In [8] a bi-linear transform was used to obtain the time-frequency plot from radar returns. The Wigner-Ville distribution was used along with a Gaussian kernel function in the bi-linear transform. The marginal distribution of frequency (MARF) was extracted for use as features in classification. A feed forward neural network (NN) was used to classify a signal as being reflected from sea clutter as opposed to a signal reflected from sea clutter and a target. The authors showed that by using a NN to classify targets an increase in accuracy of 20% was achieved compared to the CFAR method at a PFA of  $10^{-3}$ . While these results are impressive, the signals appeared to have high SCNR. In [9] ML was used to suppress sea clutter in radar returns. Doppler frequency coefficients were used as features, and both SVM and kNN classifiers were tested and compared. Only one target was present in the data used. All of these previously mentioned works used engineered features. The work presented in this paper deviates from these previous applications of ML on target detection in that it relies on ML architectures to discover descriptive features. By using this strategy the limitations of an engineered feature set, namely assumptions of the problem and domain knowledge, are avoided.

Some works in literature have presented ML methods for detecting targets in maritime clutter without the need to explicitly engineer features. Using the chaos theory of dynamic sea clutter, a NN approach was presented in [10]. The network was used to predict the next value in the time series data given  $N$  previous data points. The value of  $N$ ,

which also corresponds to the number of input nodes to the NN, was determined by computing the amount of predictability of the chaotic system. The NN had two hidden layers of size 80 and 55. A CFAR algorithm was implemented for target detection, in which the error of the model predictions were used as input to the CFAR algorithm. Both coherent and non-coherent networks were trained and tested. The results were compared to results obtained by traditional magnitude-based CFAR algorithms, and it was shown that the coherent NN method outperformed the CFAR only methods. The theory underpinning the design of the classifier was controversial, however. Unsworth *et al.* [11] and McDonald and Damini [12] showed that the methods outlined in [10] resulted in false detections for chaos. Haykin conceded in [13] that the methods discussed in [10], namely correlation dimension and Lyapunov exponent analysis, do not provide conclusive proof that a given signal is from a chaotic system, a non-linear deterministic system with added noise or a colored noise stochastic system.

A novel convolution-only CNN was presented in [14] for classification of targets in synthetic aperture radar (SAR) imagery. The presented method was two staged- first detecting targets and then classifying the target as one of ten possible types. The maximum accuracy was higher for the convolutional-only CNN than traditional CNNs (containing fully connected layers), but the probability of detection (PD) vs probability of false alarm (PFA) performance did not exceed the traditional CNNs. Many ML applications have been presented for SAR imagery (see [15], [16]). It is important to emphasize that SAR imagery is much richer in spatial information due to its higher resolution than WAS, meaning these previous studies cannot be used as benchmark results for the work in this paper.

In this paper a novel method of detecting maritime targets in WAS radar by augmenting a traditional cell averaging CFAR detector with ML is presented. ML is used in order to exploit neglected information contained within the radar data, namely spatial and temporal patterns. By choosing ML networks which are able to discover features, the process of engineering features (a process which is limited by knowledge and assumptions on the underlying data) can be avoided. The training and testing data is non-coherent (i.e. non-complex, real data only), meaning Doppler information is not available.

#### B. DEEP LEARNING SPATIAL-TEMPORAL APPROACHES

Feature sets are designed by an engineer, based on their understanding of the data set. This limits the effectiveness of the feature set by the amount of domain knowledge the engineer possesses. To avoid this limitation, it is possible to use ML networks which discover their own features. The proposed method is a combined approach, using a CNN and LSTM (an extension of the recurrent neural network (RNN)) for spatial and temporal pattern recognition respectively. Similar combination deep learning approaches have been previously proposed in other research areas [17], [18]. As ML has had limited use in radar target detection it is necessary to

**TABLE 1.** Parameters of radar.

Parameter	Value
Operating Frequency	9410 MHz
Output Power	25 kW
Antenna Length	8 ft
Antenna Beamwidth	1 deg @ -3dB
Rate of Rotation	20 RPM
Min Range	0 NM
Max Range	8 NM
Number of Range Samples	2048
Number of Azimuthal Samples	1024 (360 deg)
Amplitude Resolution	8 bit quantization

borrow concepts from other areas which have a richer history of ML application.

A combined approach, much like the one that is proposed in this work, was presented by Sainath *et al.* [19] for use in speech processing. Features were discovered by the CNN and then extracted from one of the intermediate layers of the network. The features were then reduced using a feature space reduction technique and then passed to an LSTM as input. Like the CNN, the LSTM was only used for discovering features. A fully connected NN was used for classification. The network presented in [19] provided a lot of inspiration for the work presented in this paper, however many of the details were changed to make it suitable for maritime WAS radar data. The feature reduction stage of [19] was not included. Instead, a fully connected layer in the CNN was used to reduce the size of the output (or input to the LSTM). This was done so that the feature reduction stage is incorporated into the training process of the CNN.

### C. OUTLINE OF PAPER

The remainder of the paper is organized as follows: Section II discusses the data set used for this work, Section III describes the ML architecture used, Section IV presents the results, Section V provides concluding remarks and Section VI mentions limitations of the work presented in this paper along with suggested future work.

## II. DATA SET

The data used in this work was supplied by Denbridge Marine (Birkenhead, UK) and was recorded at St. Annes Head in Pembrokeshire, Wales. A Sperry Marine Bridgemaster X-band magnetron non-coherent radar was used. The antenna rotated in the azimuth direction, with each full rotation forming a single frame of range-azimuth data. The relevant parameters of the radar are displayed in Table 1. The radar data was quantized with 8 bit resolution (non-complex values) by the receiver circuit. Due to the quantization of the data, much of the far-range low-magnitude clutter became zero in value.

Data used in this work was collected over 20 minutes, corresponding to 400 frames and contained four targets (boats). Automatic identification system (AIS) locations were not available for the 4 targets, so the locations had to be manually labeled by an operator. The data was analyzed to identify all targets present, however, it is possible that there were other

small or stationary targets present that went unlabeled. The tracks of these targets can be seen in Figure 1, while images of the targets can be seen in Figure 2. The relative magnitude and range-azimuth extent varies from one target to another. Target 1 is large and clearly visible to an untrained observer whereas the other three targets are small and weak in magnitude (i.e. small boats) when compared to surrounding clutter. Since the targets are of different sizes and have different radar cross sections, the detector performance will largely depend on the type of the target. The SCNR value of each target plotted against the frame number is shown in Figure 3. Target 2 was located far from the radar for the majority of the collection- beginning approximately 10.8 km from the radar and ending 8 km from the radar. As previously mentioned, the quantization of the data results in some data cells (where cell is defined as the range-azimuth bin of the radar returns) being zero in magnitude. SCNR is calculated by dividing the target cell value by the average of the local clutter cells  $A_{c+n}[\cdot]$ :

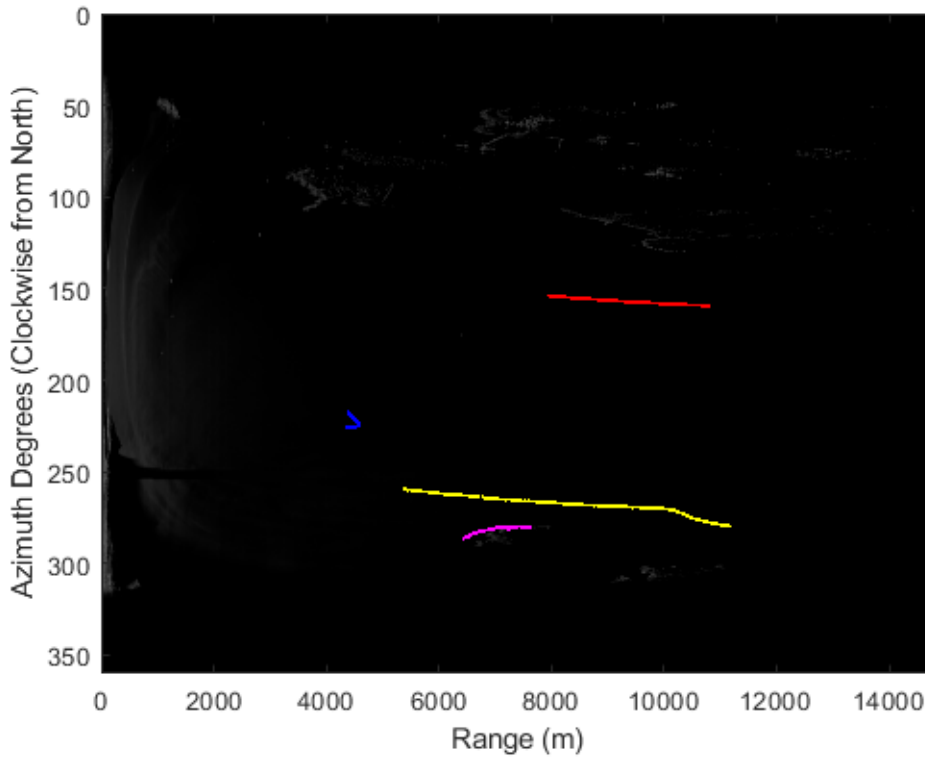
$$A_{c+n}[m, n, t] = \frac{1}{(2a + 2g + 1)^2 - (2g + 1)^2} \cdot \left( \sum_{i=m-(a+g)}^{m+(a+g)} \sum_{j=n-(a+g)}^{n+(a+g)} X[i, j, t] - \sum_{i=m-g}^{m+g} \sum_{j=n-g}^{n+g} X[i, j, t] \right) \quad (1)$$

$$SCNR[m, n, t] = \log\left(\frac{X[m, n, t]}{A_{c+n}[m, n, t]}\right) \quad (2)$$

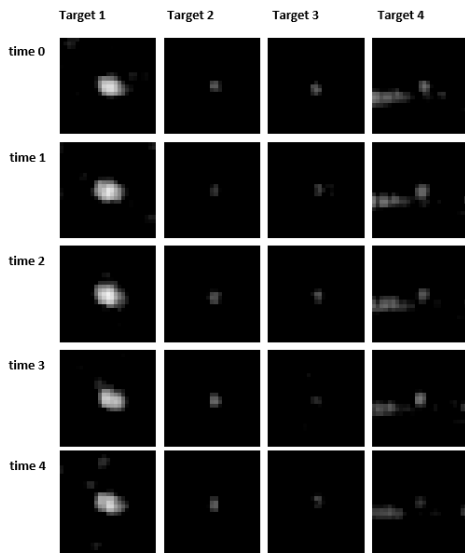
where  $m$ ,  $n$  and  $t$  are the range, azimuth and time locations of a target,  $X[\cdot]$  is the radar data,  $a$  is the width of cells taken for averaging and  $g$  is the width of guard cells around the target location. If the average of the local clutter cells is zero, the SCNR is undefined due to a division by zero. Only the SCNR values which are not undefined are shown on the plot in Figure 3. As seen in Figure 3, the SCNR fluctuates for each target over time. SCNR peaks when the target is furthest from the radar and SCNR dips the lowest when the target is closest to the radar.

## III. CNN-LSTM CLASSIFIER DESIGN

The convolutional long short-term memory neural network (CNN-LSTM) utilized in this work is composed of two parts: a CNN which is tasked with extracting spatial information within the data, and an LSTM, for analyzing temporal changes in the spatial information of the data. Finally, fully connected layers are used to make the classification. Figure 4 provides a block diagram of the proposed network. The network in this application is used to augment traditional CFAR detection methods. The first stage of the block diagram is to cell-average the data to normalize power across the range extent. The proposed method incorporates a segmentation step to limit the size of the input data to the classifier. These segments are then masked by multiplying the segments with the results of the classifier (an integer value of 0 or 1). Once the data has been masked to remove excess clutter,

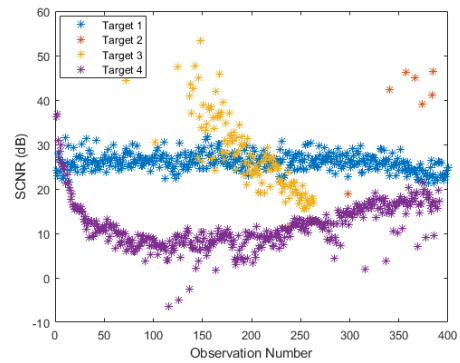


**FIGURE 1.** Target tracks superimposed on averaged frames. Target 1 in blue, Target 2 in red, Target 3 in yellow and Target 4 in purple. Y-axis indexes azimuth direction while x-axis indexes range samples. The data is collected in a rotational sweep of the radar therefore the azimuthal index corresponds to a clockwise rotation of the compass from North-East-South-West-North where each rotational sweep is a frame.



**FIGURE 2.** Example segments of frames featuring each of the four targets. Each column is five successive samples of the target.

a traditional detector structure is applied. By incorporating the ML network with a traditional CFAR detector, it could be integrated into legacy radar systems. Upon detection, a tracker is applied to further improve the performance of the algorithm. The following subsections detail each block in the block diagram in Figure 4.



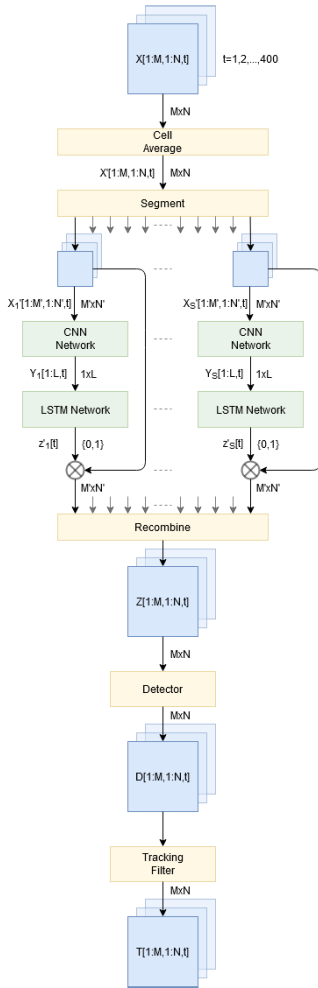
**FIGURE 3.** SCNR values for each target over duration of data collection, with  $a = 32$  and  $g = 8$ . Target 1 moves little in the radial direction, Target 2 moves towards the shore, Target 3 moves towards shore and Target 4 moves along the coast of an island, away from shore.

**A. CELL-AVERAGING**

The raw data is cell averaged according to

$$X'[i, j, t] = \frac{X[i, j, t]}{\sum_{s=j-N_G}^{j-N_G} X[i, s, t]} + \frac{X[i, j, t]}{\sum_{s=j+N_G}^{j+N_G+N_A} X[i, s, t]} \quad (3)$$

where  $N_A$  is the number of cells averaged on either side of the cell and  $N_G$  is the number of guard cells on either side of

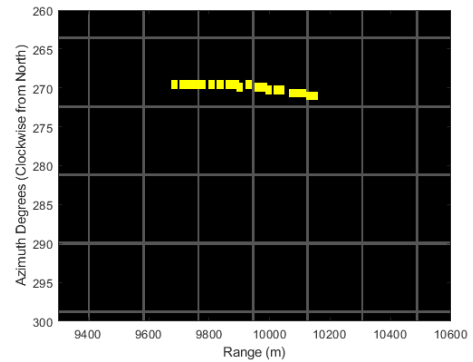


**FIGURE 4.** CNN-LSTM ML network.  $X[\cdot]$  is the radar data input,  $t$  is the time sample of  $X[\cdot]$ ,  $M$  is number of azimuthal samples,  $N$  is number of range samples,  $X_c[\cdot]$  is the cell-averaged data,  $S$  is number of segments  $X_s[\cdot]$  created by the segmentation block,  $L$  the length of the feature vector  $Y[\cdot]$ ,  $z[\cdot]$  is the binary output of the LSTM classifier,  $Z[\cdot]$  is the masked radar data,  $D[\cdot]$  is the binary valued detection tensor and  $T[\cdot]$  is the output of the tracking filter.

the cell. In this paper, the values of  $N_G = 16$  and  $N_A = 8$  are used.

### B. IMAGE SEGMENTATION

The input data is a matrix, or frame, of size  $1024 \times 2048$ , corresponding to a circular area of  $689 \text{ km}^2$ . The target signature is very small (on the order of  $10^0$  to  $10^1$  cells) when compared to the area of the entire data frame (on the order of  $10^6$  cells). Furthermore, multiple targets could be contained within any given frame. In the case of this data set, there are 4 targets present in each frame. The input frame is segmented into a grid of contiguous small segments. The benefits of segmentation are (1) reducing the size of the frames results in fewer input parameters, and (2) the targets are more likely to be isolated to individual segments, or regions of the overall data frame. A size of  $25 \times 25$  was chosen for the segmentation of the data. The size was chosen because it was large enough to fully contain the largest target



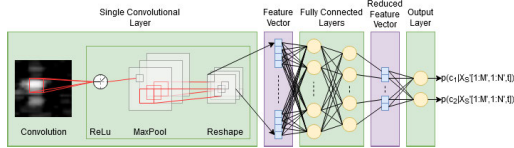
**FIGURE 5.** Example of target trajectory through adjacent segments. The grid represents the boundaries of the segments, and the yellow pixels represent locations of Target 3 over a 1 minute interval.

seen in the data set. Furthermore, the smaller the size of the segment, the better the targets' isolation from the surrounding clutter. Occasionally, targets may move into an adjacent segment which then results in the target being 'split' for a short duration. This is graphically represented in Figure 5. This could potentially cause a missed detection. For this paper, this issue is not addressed. Future work will be devoted to identifying the scope of this problem and a solution. Potential solutions could involve overlapping segmentation or using a majority rules classification for adjacent segments.

### C. CNN

CNNs are a special type of multi-layered NN. CNNs are used extensively in the field of computer vision due to their inherent ability to classify 2D images [20]. Unlike conventional NNs, CNNs are composed mainly of layers that have sparse weight connections which means there are fewer parameters to optimize. The underlying principle in CNNs is that 2D filters, which comprise a convolutional layer, are convolved with input images to create 2D feature images. These filters (convolutional layers) have adjustable weights which are optimized during the training process [20]. Convolutional layers can be stacked in series to derive high order features. The result of training is that specialized features are discovered by the network. Each convolutional layer has architectural parameters which must be defined by the user. These architectural parameters include the size of the filters, the number of filters, amount of padding to the input and filter stride length. Additionally, the number of convolutional layers must also be defined. CNNs include other layer types such as pooling and activations which are explained in detail in [20].

CNNs can be considered as consisting of two stages: feature extraction via convolutional layers and classification via fully connected layers. This is shown in Figure 6. When multiple fully connected layers are used, the size of the feature vector changes according to the number of nodes in each layer. Fully connected layers can therefore be used to reduce the size, or dimensionality of the feature vector before it is passed on to the next stage, the LSTM. The CNN is trained,



**FIGURE 6. Graphical example of CNN with single convolutional layer.** Here  $p(c_1|X'_S[1 : M', 1 : N', t])$  and  $p(c_2|X'_S[1 : M', 1 : N', t])$  represent the posterior probabilities of class  $c_1$  and  $c_2$  given the input of  $X'_S[1 : M', 1 : N', t]$ .

and then the final fully connected layer is removed so that the output of the CNN becomes the input to the next stage of the CNN-LSTM classifier. The number of convolutional and fully connected layers, as well as the number of filters or nodes in each layer, is determined through trial and error, with the goal of maximizing test set accuracy and minimizing test set false alarms.

An important aspect of the CNN is its ability to discover features rather than requiring the user to engineer features. This makes CNNs useful for problems in which the model for the underlying system dynamics are unknown or too complex. Rather than developing custom design features based on domain knowledge by users, the task is left to the CNN to discover an optimal feature set based on spatial patterns. Furthermore, targets move and therefore shift in space over time. Therefore, it is preferable to use a classifier which is invariant to translations (CNNs are invariant to shift translations due to convolutional and max-pooling operations [21]). Automatic feature discovery can also be achieved through other methods such as auto-encoders or k-means clustering.

Many specialized CNN architectures have been presented in literature, such as AlexNet [22], VGG-16 [23], Inception-V1 [24] and ResNet-50 [25] which show marked improvement over conventional CNNs. These networks are very deep and contain millions of parameters. These networks were designed for high resolution images, which is not what this paper deals with. These networks are therefore not considered as candidates for the CNN in the CNN-LSTM network presented in this paper.

The input to the CNN-LSTM classifier is a sequence of segments  $X'_S[1 : M', 1 : N', t]$ . Unlike the LSTM, the CNN has no time dependency and therefore does not exploit temporal patterns.

**D. LSTM**

LSTM is an extended form of the RNN. LSTMs were proposed as a solution to the vanishing gradient problem of RNNs [26]. Furthermore LSTMs can capture both long-term and short-term temporal patterns within the data unlike RNNs. Many versions of LSTMs exist (see [27] for an excellent review of these), however the basic LSTM consists of a block input, an input gate, a forget gate and an output gate. Each of these gates take as input the data vector  $Y[1 : L, t]$  and the recursive output  $z'[t - 1]$ . The purpose of the gates within the LSTM is to control the flow of information from the input to output while maintaining a memory of past inputs

(cell state). It is this basic form of the LSTM that will be used in this paper. The output of the LSTM,  $z'[\cdot]$ , is the learned posterior probability that the segment belongs to the clutter-only class.

**E. FULLY CONNECTED LAYERS**

The data is sparsely populated with targets and therefore, has a large class imbalance. Conventional or default loss functions for fully connected neural networks (or any binary classifiers for that matter) assign equivalent loss to misclassification of both classes. For highly imbalanced data sets, these classifiers can achieve very low loss and high overall accuracy by simply classifying the over-represented class very well and sacrificing the classification performance of the under-represented class. In the extreme case, this would mean classifying all observations as belonging to the over-represented class. For the application considered in this paper, this would result in all clutter being masked along with all targets. In order to remedy this, a cost or weight term for incorrectly classifying each class is added to the cost function. In particular, data which is over represented would have a weight of less than 0.5 while the under-represented class would have a weight greater than 0.5 (assuming normalized weights). The higher the weight, the more the classifier favors correct classification for that class, albeit at the expense of the other class. The two extremes of this weighted loss function are when both weights are 0.5 (which simplifies to the original unweighted function) and when the weights are 1 and 0, it will likely assign a single class label for all data points. The weighted cost function is defined as follows

$$L = -\frac{1}{l} \sum_{s=1}^l \sum_{c=1}^C w_c h_{sc} \log(z'_{sc}) \tag{4}$$

where  $l$  is the number of data points in the training set,  $C$  is the number of classes (2 in this work),  $h$  is the truth value and  $z'$  is the output of the classifier. A weighted cost function is used for the training of both the LSTM and the CNN. The actual values of the weight  $w_c$  are computed for each network based on the ratio of training samples for each.

**F. SEGMENT MASKING**

Segments are masked based on the result of the LSTM classifier:

$$X'_1[\cdot] = \begin{cases} 0 & z'_1[\cdot] > CThresh \\ X'_1[\cdot] & otherwise \end{cases} \tag{5}$$

where  $CThresh$  is a value between 0 and 1. The value  $CThresh$  defines the operating point of the classifier, it represents how much confidence is placed on the classifier output. The higher the value of  $CThresh$ , the less confidence is placed on the classifier. When  $CThresh = 1$ , the classifier does not have any effect on the flow of data therefore the results are equivalent to baseline.

After segment masking, the segments are reassembled to form the original frame (albeit masked according to the chosen  $CThresh$  value).

### G. DETECTOR STRUCTURE

The detector structure in this work is the threshold-based cell averaged (CA) CFAR [3], which defines any point greater or equal to the set threshold as a detection. Because the cell under test (CUT) is scaled by the surrounding clutter power, the threshold can be considered as being adaptive. The output of the detector is a binary valued multidimensional array of size  $M \times N \times m$ , where  $m$  is the number of frames in the data set. The output of the detector is defined as

$$D[i, j, t] = \begin{cases} 1 & Z[i, j, t] \geq Thresh \\ 0 & otherwise \end{cases} \quad (6)$$

where  $Z[i, j, t]$  is the cell averaged data with the segment masking applied.

### H. TRACKING FILTER

In typical operation, a tracking filter is applied to detections to build target tracks [28]. Trackers make use of association rules to cluster nearby detections and also to associate detections over a series of observations to form tracks. Trackers make assumptions about the velocity characteristics of targets in order to determine if successive detections are generated from a moving target or generated from clutter. Tracking filters are effective at suppressing false detections and improving detection performance.

The tracker used in this study used a global nearest neighbor (GNN) algorithm [28]. Tracks were confirmed if 4 of the past 5 steps had detections, and deleted if less than 9 detections occur in the previous 10 steps. These parameters were found through trial and error to provide the lowest number of false tracks while maintaining the maximum detection rate. The filter uses a near constant velocity (NCV) Kalman filter [28]. Prior to applying the tracker the detections are clustered using a connectivity-based clustering algorithm [29] with a maximum Euclidean distance of 15. This is done to collapse multiple detections from a single reflector to one detection. Cluster centers are taken as the new detection points.

## IV. RESULTS

### A. TRAINING

All work was done in Matlab R2020a using the deep learning toolbox. The code was run on a laptop computer with an N3700 Intel Pentium CPU (1.60 GHz) and 8 GB of RAM. Table 2 shows the parameters used in the network. The parameters were found mainly through trial and error. Training occurred in the following order:

- 1) The CNN was trained with the custom weighted cost function. The training set consisted of  $25 \times 25$  cell segments.
- 2) The final fully connected layer was stripped from the CNN.

TABLE 2. Parameters of classifier network.

Parameter/ Layer	Value
Segment Size	25 x 25
Train-Test Partition	50%
CNN Filter Size	3 x 3
CNN Convolutional Layer 1	32 filters
CNN Convolutional Layer 2	32 filters
CNN Convolutional Layer 3	32 filters
CNN Fully Connected Hidden Layer 1	32 nodes
CNN Fully Connected Hidden Layer 2	32 nodes
CNN Fully Connected Hidden Layer 3	32 nodes
CNN Final Fully Connected Layer	2 nodes
CNN Class Weights	[0.3663, 0.6337]
LSTM Recurrent Hidden Layer	32 nodes
LSTM Fully Connected Hidden Layer 1	32 nodes
LSTM Fully Connected Hidden Layer 2	32 nodes
LSTM Final Fully Connected Layer	2 nodes
LSTM Class Weights	[0.1059, 0.8941]

- 3) The LSTM was trained with the custom weighted cost function. The training set consisted of sequences of  $25 \times 25$  segments which had been passed through the CNN.

An Adam optimizer [30] was used for training the CNN and LSTM.

### B. CNN TRAINING SET

Data from the first 50% of frames were used for training and the final 50% was reserved for testing. The target class was sampled with some overlap so that the number of samples could be increased for training. This was performed by taking 16 segments for each of the four targets from each of the first 200 frames (with the exception of Target 3 which is not present in the first 128 frames of the data set). This resulted in a total of 10752 segments associated with a target. The overlap between target samples was 5 cells in the range and azimuth axes. For the non-target class, 18600 segments (not containing a target) were taken at random from the first 200 frames. The ratio of non-target to target class samples was 18600 to 10752. The cost function weights were found to result in the best PD to PFA performance when they were approximately equal to the relative proportions of the class imbalance, therefore the weights of the cost function for the CNN,  $w_c$ , were set to [0.3663, 0.6337]. Training took 33 minutes and 8 seconds to complete 5 epochs with a mini-batch size of 512 (57 iterations per epoch).

### C. LSTM TRAINING SET

The training-testing split of the data set was the same for the LSTM and the CNN but the sampling method was different as LSTM operates on sequences of segments. As a result, storage requirements for training and testing will be very large. Therefore, it was not practical to use the same over-sampling method used for training the CNN. For the target class, 25 sequences of segments were randomly selected for each of the four targets. Each of these sequences contained a target in at least one segment. An additional 220 sequences which had no targets present in any segment were randomly

selected. The total number of sequences was 320, meaning a total number of segments of 64000. The ratio of segments not containing a target to containing a target was 57222 to 6778, therefore the weights of the cost function for the LSTM,  $w_c$ , were set to [0.1059, 0.8941]. Training took 25 minutes and 46 seconds to complete 500 epochs with full batches.

**D. PERFORMANCE METRICS**

Detectors are evaluated based on their PD vs PFA rates. For this work, PD is defined as the percentage of frames in which a particular target is detected. Detectors operate on each cell of the frame whereas targets cover many cells. A positive detection,  $Pd$ , occurs when a positive finite number of detections greater than 0 are made within a small area centered on a target:

$$Pd_o[t] = \begin{cases} 1, & \sum_{i=L_{o1}-I}^{L_{o1}+I} \sum_{j=L_{o2}-J}^{L_{o2}+J} D[i, j, t] > 0 \\ 0, & \text{otherwise} \end{cases} \quad (7)$$

where  $L_{o1}$  and  $L_{o2}$  is the location of target  $o$  in the first and second axes respectively (range and azimuth) and  $I$  and  $J$  are the margins of the target area in both axes. The margins are used so that small errors in the labeled points  $L_{o1}$  and  $L_{o2}$  do not effect the computation of the PD. In this paper, the margins were set to  $I = 10$  and  $J = 10$ . PD is then computed by simply taking the average of the positive detections  $Pd_o$  for each target:

$$PD_o = \frac{1}{m} \sum_{t=1}^m Pd_o[t] \quad (8)$$

where  $m$  is the total number of time samples  $t$  used in the computation of the average. A false detection, or false alarm is defined as

$$Fd_{i,j}[t] = \begin{cases} 0, & \text{if } L_{o1} - I \leq i \leq L_{o1} + I, \\ & L_{o2} - J \leq j \leq L_{o2} + J, \\ & \forall o \in [1, 0] \\ D[i, j, t], & \text{otherwise} \end{cases} \quad (9)$$

which means any detections outside of the regions labeled as targets are considered false alarms. The false alarm rate is computed as the number of false detections divided by the maximum possible number of false detections, as defined by

$$PFA = \frac{\sum_{t=1}^m \sum_{i=1}^M \sum_{j=1}^N Fd_{i,j}[t]}{(M \cdot N \cdot m) - (4 \cdot I \cdot J \cdot m)} \quad (10)$$

where the number 4 is the number of targets in this particular data set. ROC curves are constructed by computing  $PD_o$  and  $PFA$  in a threshold sweep. The PD and PFA calculations use apriori information about the targets, which means they require labeled data.

**E. DEFINITION OF BASELINE**

In this paper, baseline is defined as the results obtained from the cell averaged data without undergoing ML augmentation. This is equivalent to when  $CThresh$  is set to 1. All performance metrics are computed the same way for the baseline and ML approach.

**F. ANALYSIS OF RESULTS**

The ROC curves as seen in Figure 7 are the results of applying the network architecture from Figure 4 to the test data. The ROC curves are plotted against the  $\log_{10}(PFA)$ . Seven different operating points were tested as can be seen in Figure 7. The value of  $CThresh$  for each operating point can be found in the legend. As previously stated,  $CThresh = 1$  is equivalent to the baseline approach. The effect that ML has on the ROC curves is a shift towards lower PFA. The bend in the averaged ROC curve occurs at approximately  $10^{-2.7}$  to  $10^{-3}$  for baseline, whereas it occurs between  $10^{-3.3}$  to  $10^{-3.6}$  in the ML augmented curves. The cost of lower PFA, however, is a reduction in PD. As the operating point is adjusted to more blanking (lower  $CThresh$  value), the PFA drops along with PD. The use of the operating point parameter  $CThresh$  would give an operator the control over PFA and PD performance. The performance is improved overall, with significant improvement for Targets 2 and 4. The performance remains unchanged for Target 1, likely because it is detectable at very low PFA to begin with. Target 3 is the only target for which the performance doesn't improve in both PFA and PD. The PFA is lowered for Target 3, but the PD remains limited even for the operating point of  $CThresh = 0.9$ . It is speculated that the performance is the least improved for Target 3 because of all the targets it moves across the range extent much faster than the rest. This means that the target is present in any particular segment for a much shorter period than for the other targets. This restricts the amount of temporal information that the LSTM can exploit for identifying this target.

Figure 8 shows an example of the clutter masking capability of the CNN-LSTM network. On the left pane of Figure 8, a region of a single frame containing Target 1 is shown using the baseline approach, while the right pane of the figure the same region of the same frame but with masking applied with  $CThresh = 0.2$  is shown.

A sliding window was applied over the baseline data to average each frame with the frames one time sample behind and ahead (three frames averaged). The sliding window filter degraded the performance, therefore the resulting ROC curves are not included in this paper. The performance degradation is speculated to be due to the large time between samples causing the targets to de-correlate between successive frames. Each range cell is approximately 7.23 m in length, and sampling is performed at a rate on 1 frame every 3 seconds. The fastest target travels radially at 7.13 m/s meaning for each successive sample it covers a distance equal to approximately 3 range cells.



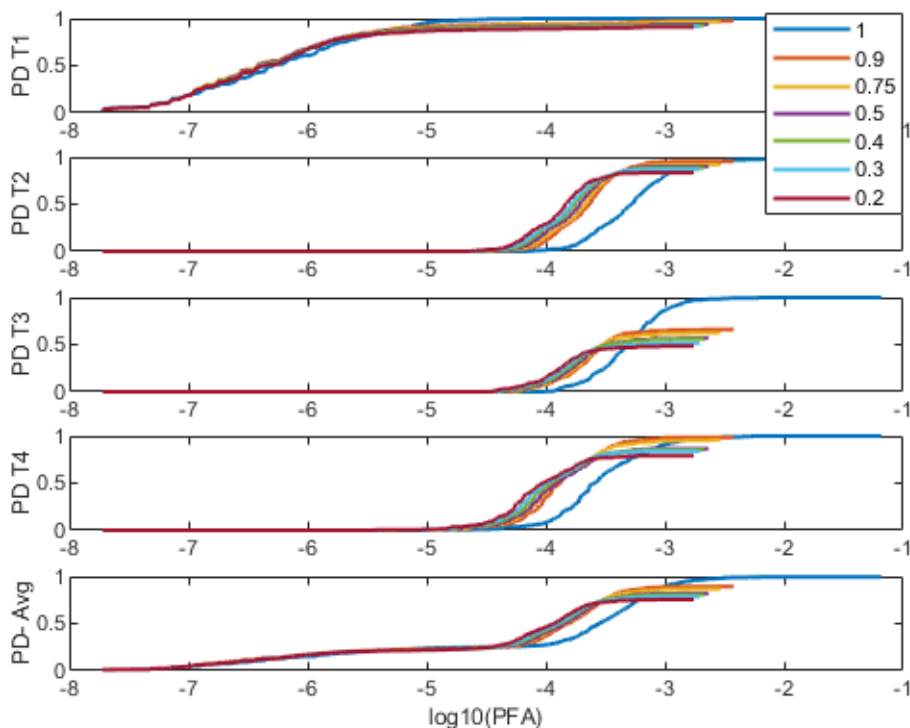


FIGURE 7. ROC curves for validation data. The legend shows the corresponding values of  $C_{Thresh}$  used.

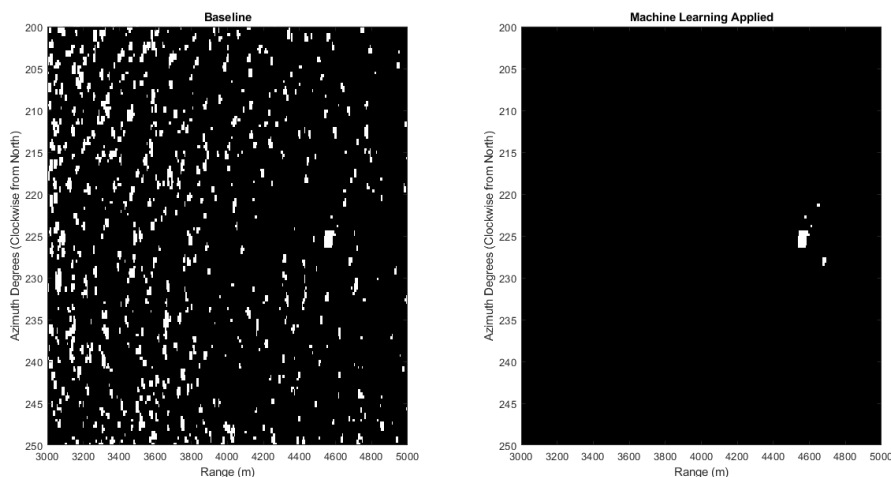


FIGURE 8. Example of a region of a frame without (left) and with (right) machine learning applied. Target 1 can be seen near the point (4600, 225) in both images. A value of  $C_{Thresh} = 0.2$  was used for masking.

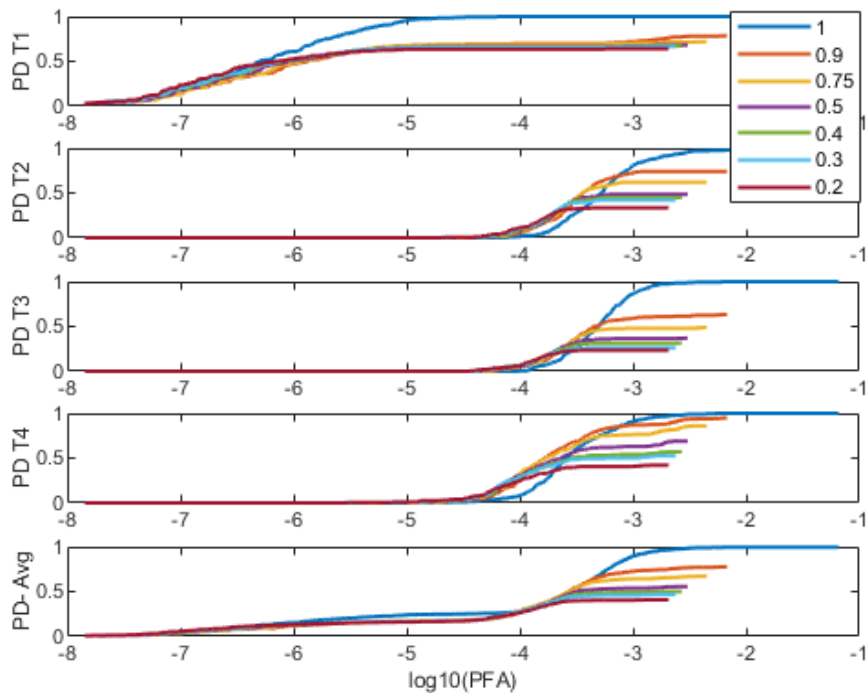
**G. CNN ONLY**

To verify that the LSTM portion of the algorithm contributed positively to the performance, ROCs were generated by using only the output of the final layer of the CNN, effectively removing the LSTM. These results can be seen in Figure 9. An interesting result of this test is that the CNN alone was inadequate in classifying the data, but the CNN-LSTM combined approach performed better than the baseline. A possible reason for this is that spatial information alone is not enough

to identify targets within a field of clutter. This is evident when looking at a single frame of the data without the context of time. Targets can be very difficult to identify even by direct observation without observing multiple frames over a short period of time.

**H. ROTATIONAL INVARIANCE**

The frames of the test set were rotated 90° clockwise and fed to the trained CNN-LSTM classifier to test the rotational



**FIGURE 9.** ROC curves for each target using only CNN. The legend shows the corresponding values of  $C_{Thresh}$  used.

invariance of the classifier. The resulting ROC curves can be seen in Figure 10. The performance degradation was negligible for Targets 2 and 3. This suggests that the performance reduction was derived from reduced detectability as opposed to increased false alarms. If there was an increase in false alarms it would be expected that there would be a performance degradation associated with all target ROC curves. There was, however, only two target ROC curve showing any significant change in performance. Figure 2 helps lend insight into this result. Target 1 has a well defined shape, and when the target is rotated it possibly appears novel. Unlike Targets 1-3, Target 4 was surrounded by ground clutter returns associated with coast lines. It is speculated that the network discovered features that were consistent with the shape and orientation of the coast line near Target 4 as opposed to features related to the target. There was no rule explicit in the training of the classifier to prevent it from discovering such features. Rotating the coast line to align in a vertical direction possibly caused the target to appear novel. This would indicate that the network discovered some non-useful spatial features which would lead to poor generalization. This insight shows the need for large target-rich data sets, and even target-free coastal data, for development of robust ML-based detectors with good generalization performance. One mitigating technique for this issue would be to augment the training set with rotated copies of the original frames. Another option may be to implement a coast line masking classifier.

### I. SPATIAL FEATURES

Aim 1 of this paper is to determine if the features discovered by the CNN were based on spatial patterns. By destroying

the spatial patterns of the segments insights could be gained into the dependence of the CNN-LSTM network on the CNN portion of the network, since this is the portion that relies solely on spatial information. To this end, each segment was shuffled by randomizing the elements' indices. This removed spatial correlations while preserving magnitude characteristics. Figure 11 shows the results of this test. Detectability of Target 1 was essentially unchanged. This is possibly because Target 1 is high in intensity compared to surrounding clutter, as shown in Figure 3, and so the classifier is relying solely on the amplitude statistics. Detectability of Target 4 was compromised as was expected due to the fact that it exhibited distinct spatial signatures (i.e. spatial distribution in space) and low SCNR. An unexpected result was that the ROC performance curves for Target 2 and 3 showed that detectability was improved at a lower PFA compared to baseline. It is speculated that this was not because of an increased ability to detect these targets after they had been shuffled, but rather the clutter became less 'target-like' after it had been shuffled. Certain regions of the frame contained land clutter which had persistent and well defined shapes that may have appeared similar to targets. When the segments corresponding to these regions were shuffled, they appeared more like clutter than targets. A possible reason that the detectability of these two targets was preserved even after randomly shuffling the indices of the segments was because the targets contained little spatial information/structure to begin with. The magnitude statistics over time however, were completely preserved, which would still be identified by the LSTM nodes. This shows that certain targets may be identified by spatial features whereas others are not.

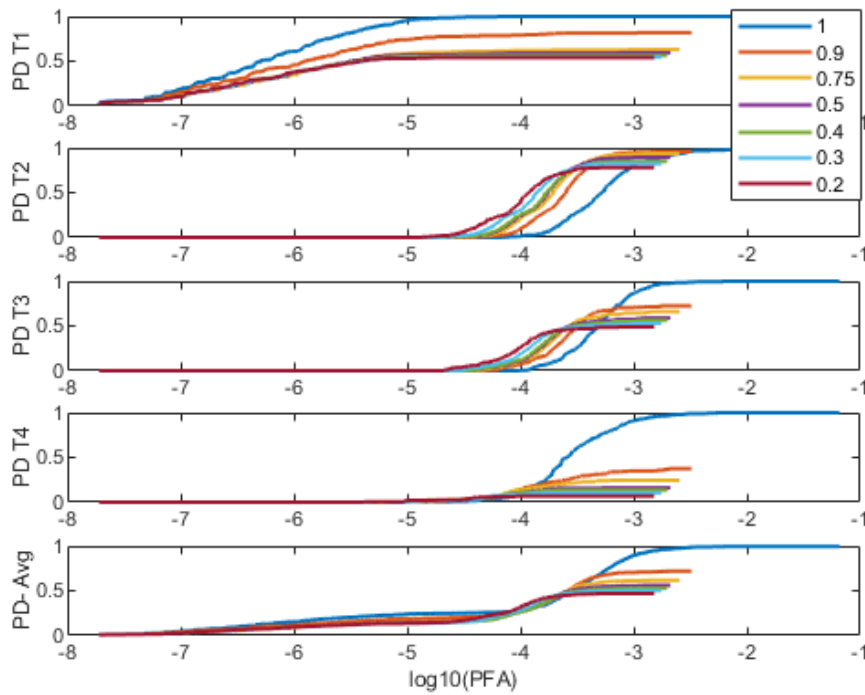


FIGURE 10. ROC curves for each target with and without frame rotation. The legend shows the corresponding values of  $C_{Thresh}$  used.

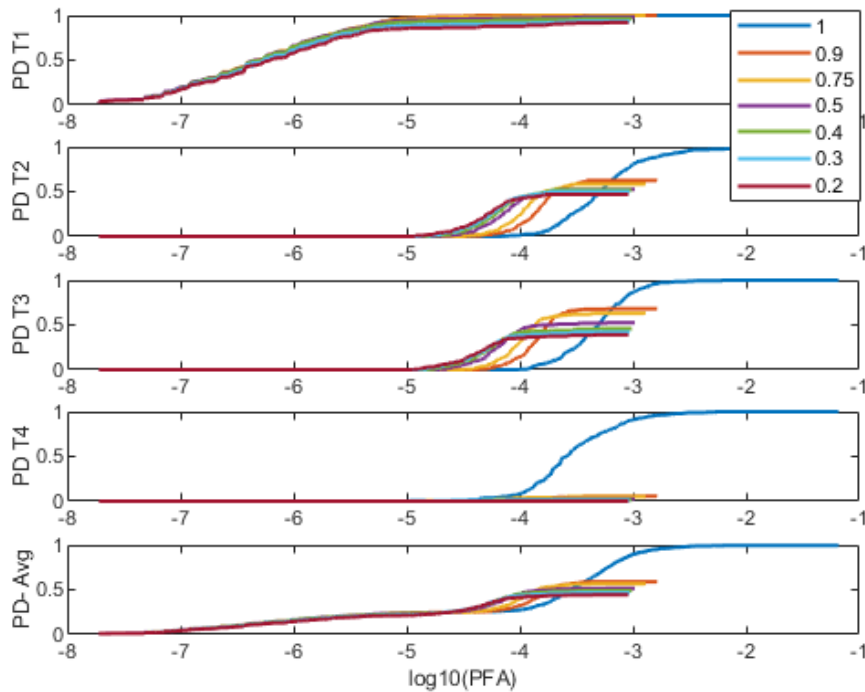


FIGURE 11. ROC curves for each target with space shuffling. The legend shows the corresponding values of  $C_{Thresh}$  used.

### J. TEMPORAL PATTERN RECOGNITION

To test the effectiveness of the LSTM recurrent nodes in their ability to exploit temporal patterns in the data (as per aim 2 of the paper), the sequences of radar frames in the test set were shuffled to randomize the order of the frames. The effect of doing this was to destroy any temporal patterns existing in the

data. The resulting ROC curves for the CNN-LSTM can be seen in Figure 12.

As might be expected, the overall performance of the classifier dropped significantly when the order of the frames were randomized. Target 1 was mostly detectable when the frames were shuffled. Even though successive frames were

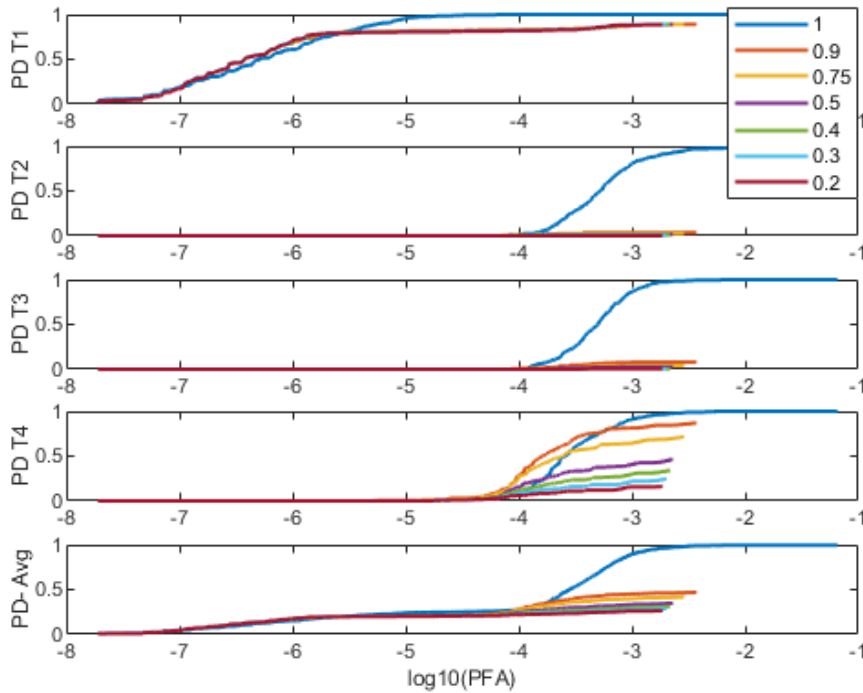


FIGURE 12. ROC curves for each target with time shuffling. The legend shows the corresponding values of  $C_{Thresh}$  used.

randomized, it is speculated that the relatively slow motion of Target 1 along with its well defined and consistent shape allowed detections to be made. Similarly, Target 4 moved relatively slow and was surrounded by well defined land clutter shapes. It is possible that the classifier was able to still identify this target for the same reasons as for Target 1. Targets 2 and 3 were undetectable in the randomized frames. This showed that the network relied strongly on the temporal changes of these targets in order to classify them correctly. This observation is consistent with intuition, given the example segments of the targets in Figure 2. Targets 2 and 3 had little to no spatial features that allowed them to be contrasted against clutter. Furthermore, Targets 2 and 3 moved much faster and further than Targets 1 and 4 meaning that when randomized, successive segments were even less likely to retain any temporal correlation of those targets.

Overall, the results from this trial show the ability of the LSTM recurrent nodes specifically, in exploiting temporal features.

### K. TRACKING

The tracker, as described in Section III-H, was applied to both the baseline data and the ML-masked data from Section F in the Results. Detections made at a fixed PD of 0.7 with  $C_{Thresh} = 0.2$  were used for the tracking stage. The results of the tracking algorithm can be seen in Table 3, where  $N_{Tracks}$  was the number of confirmed tracks for each trial.

With ML, the average PD was decreased from 0.9325 to 0.7700. The false alarm rate however, was reduced significantly with the ML augmented detections, from  $10^{-4.1298}$

TABLE 3. Performance metrics of tracker.

Metric	Baseline	ML Augmented
$PD_1$	1	0.8950
$PD_2$	0.9450	0.8550
$PD_3$	0.8400	0.5350
$PD_4$	0.9450	0.7950
$PD_{AVE}$	0.9325	0.7700
$PFA$	$10^{-4.1298}$	$10^{-4.7411}$
$N_{Tracks}$	9999	1407

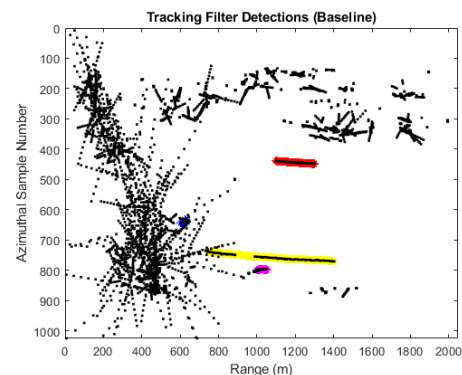
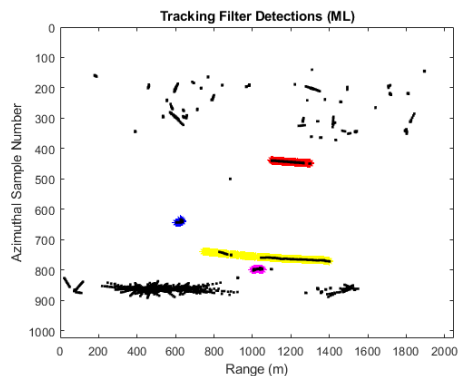


FIGURE 13. Baseline tracks with truth data highlighted (Target 1-4 highlighted in the colors blue, red, yellow and purple respectively).

to  $10^{-4.7411}$ . Furthermore, fewer tracks were generated when ML was used (1407 versus 9999). This shows that the ML-based approach presented in this paper was exploiting information that was not utilized by tracking filters. Plots of the track updates after filtering can be seen in Figures 13 and 14.



**FIGURE 14.** ML tracks with truth data highlighted (Target 1-4 highlighted in the colors blue, red, yellow and purple respectively).

## V. CONCLUSION

The purpose of this paper was to determine whether or not ML could be used as an effective means for augmenting traditional detector structures for targets in maritime environments by exploiting spatio-temporal features within the data. A CNN-LSTM approach was presented so that spatio-temporal features could be exploited for classification of target containing regions. Not only do these network architectures exploit spatio-temporal features for classification, they also discover them, bypassing the need to define custom feature sets.

It was shown that the relative importance of spatial and temporal information depended on the target characteristics, such as size, shape, velocity and surrounding environment. It was shown that the network was relatively invariant to rotations of the data. By shuffling the spatial information in the input to the network, it was shown that performance dropped due to the elimination of discovered spatial patterns. Shuffling the order of the frames resulted in a degradation of performance for fast moving targets. It was shown that the ML approach performed better than baseline, with the ROC detection curves being shifted to lower PFA values. Finally, a tracker algorithm was applied to both ML and baseline detections, and it was found that a lower PFA was achievable with ML. This showed that some of the gains achieved using ML were propagated even through the tracker filter.

## VI. LIMITATIONS AND FUTURE WORKS

While the results presented in this paper have demonstrated an ability to exploit spatio-temporal information contained within maritime radar data, initial tests done on different data sets showed poor generalized performance. Improving generalizability across different data sets is the next step in this research. Certain key future works include collecting more data so that the network depth can be increased without compromising generalizability and experimenting with different classifier models at the back end to replace the fully connected layers such as SVMs. To get the results of the classifier for the test set takes a total of 10.4 seconds per frame of data. The radar antenna rotates at 20 RPM, meaning frames are recorded every 3 seconds. The algorithm

in this paper is too slow for real time operation at the rate that it currently takes. This should be easy to mitigate by implementing changes such as coding for parallel processing, using a GPU or rewriting in a coding language that has less overhead. This paper does not address the issue of targets migrating into adjacent segments causing missed detections during the transition period. This is another issue that should be addressed in future work. Possible solutions would be to test overlapping segments, or to try and eliminate the segmentation stage by using a regression type network instead of classification type.

## ACKNOWLEDGMENT

The authors would like to thank Denbridge Marine Ltd., U.K., for supplying the data set used in this work.

## REFERENCES

- [1] S. Panagopoulos and J. J. Soraghan, "Small-target detection in sea clutter," *IEEE Trans. Geosci. Remote Sens.*, vol. 42, no. 7, pp. 1355–1361, Jul. 2004.
- [2] Anonymous, "Drug smugglers on radar," *Signal*, vol. 64, no. 7, Mar. 2010.
- [3] I. Skolnik, *Introduction to Radar Systems*, 2nd ed. New York, NY, USA: McGraw-Hill, 1980.
- [4] W. Baldygo, R. Brown, M. Wicks, P. Antonik, G. Capraro, and L. Hennington, "Artificial intelligence applications to constant false alarm rate (CFAR) processing," in *Proc. Rec. IEEE Nat. Radar Conf.*, Apr. 1993, pp. 275–280.
- [5] C. Kabakchiev, V. Behar, I. Garvanov, D. Kabakchieva, and H. Rohling, "Detection, parametric imaging and classification of very small marine targets emerged in heavy sea clutter utilizing GPS-based forward scattering radar," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2014, pp. 793–797.
- [6] N. del-Rey-Maestre, M.-P. Jarabo-Amores, D. Mata-Moya, J.-L. Barceña-Humanes, and P. G. del Hoyo, "Machine learning techniques for coherent CFAR detection based on statistical modeling of UHF passive ground clutter," *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 1, pp. 104–118, Feb. 2018.
- [7] M. K. McDonald, "Discrimination of human targets for radar surveillance via micro-Doppler characteristics," *IET Radar, Sonar Navigat.*, vol. 9, no. 9, pp. 1171–1180, Dec. 2015.
- [8] Z. Cai, M. Zhang, J. Wang, J. Chen, and J. Zhao, "A reassigned bilinear transformation and Gaussian kernel function-based approach for detecting weak targets in sea clutter," *Int. J. Remote Sens.*, vol. 37, no. 23, pp. 5668–5686, Dec. 2016.
- [9] D. Callaghan, J. Burger, and A. K. Mishra, "A machine learning approach to radar sea clutter suppression," in *Proc. IEEE Radar Conf. (RadarConf)*, May 2017, pp. 1222–1227.
- [10] L. X. Bo, "Detection of signals in chaos," Ph.D. dissertation, McMaster Univ., Hamilton, ON, Canada, 1996.
- [11] C. P. Unsworth, M. R. Cowper, S. McLaughlin, and B. Mulgrew, "Re-examining the nature of radar sea clutter," *IEE Proc. Radar, Sonar Navigat.*, vol. 149, no. 3, pp. 105–114, Jun. 2002.
- [12] M. McDonald and A. Damini, "Limitations of nonlinear chaotic dynamics in predicting sea clutter returns," *IEE Proc. Radar, Sonar Navigat.*, vol. 151, no. 2, pp. 105–113, Apr. 2004.
- [13] S. Haykin, R. Bakker, and B. W. Currie, "Uncovering nonlinear dynamics—The case study of sea clutter," *Proc. IEEE*, vol. 90, no. 5, pp. 860–881, May 2002.
- [14] S. Chen, H. Wang, F. Xu, and Y.-Q. Jin, "Target classification using the deep convolutional networks for SAR images," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 8, pp. 4806–4817, Aug. 2016.
- [15] D. Guan, X. Tang, L. Wang, and J. Zhang, "Extreme learning machine with superpixel-guided composite kernels for SAR image classification," *IEICE Trans. Inf. Syst.*, vol. E101.D, no. 6, pp. 1703–1706, 2018.
- [16] J. Geng, H. Wang, J. Fan, and X. Ma, "Deep supervised and contractive neural network for SAR image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 4, pp. 2442–2459, Apr. 2017.
- [17] Z. Zuo, B. Shuai, G. Wang, X. Liu, X. Wang, B. Wang, and Y. Chen, "Convolutional recurrent neural networks: Learning spatial dependencies for image representation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2015, pp. 18–26.

- [18] L. Deng and J. Platt, "Ensemble deep learning for speech recognition," in *Proc. Interspeech*, Sep. 2014.
- [19] T. N. Sainath, O. Vinyals, A. Senior, and H. Sak, "Convolutional, long short-term memory, fully connected deep neural networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2015, pp. 4580–4584.
- [20] A. Ng. (Dec. 2018). *Convolutional Neural Networks*. Coursera: Deep Learning Specialization. Accessed: Feb. 28, 2019. [Online]. Available: <https://www.coursera.org/learn/convolutional-neural-networks/lecture/4THzO/transfer-learning>
- [21] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016. [Online]. Available: <http://www.deeplearningbook.org>
- [22] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017.
- [23] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*. [Online]. Available: <https://arxiv.org/abs/1409.1556>
- [24] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," *CoRR*, vol. abs/1409.4842, 2014. [Online]. Available: <http://arxiv.org/abs/1409.4842>
- [25] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *CoRR*, vol. abs/1512.03385, 2015. [Online]. Available: <http://arxiv.org/abs/1512.03385>
- [26] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, May 2015.
- [27] K. Greff, R. K. Srivastava, J. Koutnik, B. R. Steunebrink, and J. Schmidhuber, "LSTM: A search space odyssey," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 10, pp. 2222–2232, Oct. 2017.
- [28] M. Betke and W. Zheng, *Data Association for Multi-Object Visual Tracking*. San Rafael, CA, USA: Morgan & Claypool, 2017.
- [29] R. Duda, P. Hart, and D. Stork, *Pattern Classification*. New York, NY, USA: Wiley, 2001.
- [30] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Representations*, San Diego, CA, USA, 2015.



**ZACHARY BAIRD** (Student Member, IEEE) received the B.Sc. degree in electrical and computer engineering from the University of New Brunswick, Fredericton, NB, Canada, in 2015, and the M.A.Sc. degree in systems and computer engineering from Carleton University, Ottawa, ON, Canada, where he is currently pursuing the Ph.D. degree with the Systems and Computer Engineering Department. His thesis focused on machine learning applications for improving maritime wide area surveillance radar detectors. He is also a Student Research Assistant with the Radar Surveillance and Exploitation Section, Defence Research and Development Canada, Ottawa.



**MICHAEL K. MCDONALD** received the B.Sc. degree (Hons.) in applied geophysics and the M.Sc. degree in electrical engineering from Queen's University, Kingston, ON, Canada, in 1986 and 1990, respectively, and the Ph.D. degree in physics from the University of Western Ontario, London, ON, in 1997. He was with the Space Science and Satellite Communications Departments, ComDev, Cambridge, ON, from 1989 to 1992. He held a postdoctoral position with the Physics Department, SUNY at Stony Brook, from 1996 to 1998. He is currently a Defence Scientist with the Radar Surveillance and Exploitation Section, Defence Research and Development Canada, Ottawa, ON. His current research interests include application of STAP processing, nonlinear filtering to the detection of small maritime and land targets, tracking and fusion, and machine learning for wide area surveillance.



**SREERAMAN RAJAN** (Senior Member, IEEE) received the B.E. degree in electronics and communications from Bharathiyar University, Coimbatore, India, in 1987, the M.Sc. degree in electrical engineering from Tulane University, New Orleans, LA, USA, in 1992, and the Ph.D. degree in electrical engineering from the University of New Brunswick, Fredericton, NB, Canada, in 2004.

From 1986 to 1990, he was a Scientific Officer with the Reactor Control Division, Bhabha Atomic Research Center (BARC), Mumbai, India. He was undergone an Intense Training in nuclear science and engineering with the Training School. At BARC, he developed systems for control, safety, and regulation of nuclear research and power reactors. From 1997 to 1998, he carried out research through a grant with Siemens Corporate Research, Princeton, NJ, USA. From 1999 to 2000, he was with JDS Uniphase, Ottawa, ON, Canada, where he worked on optical components and development of signal processing algorithms for advanced fiber optic modules. From 2000 to 2003, he was with Ceyba Corporation, Ottawa, where he developed channel monitoring, dynamic equalization, and optical power control solutions for advanced ultralong haul and long haul fiber optic communication systems. In 2004, he was with Biopeak Corporation, where he developed signal processing algorithms for non-invasive medical devices. From December 2004 to June 2015, he was a Defence Scientist with the Defence Research and Development Canada, Ottawa. He was an Adjunct Professor with the School of Electrical Engineering and Computer Science, University of Ottawa, Ottawa, from July 2010 to June 2018. He joined the Department of Systems and Computer Engineering, Carleton University, as a Tier 2 Canada Research Chair (Sensors Systems) in July 2015. He has been an Associate Director with the Ottawa-Carleton Institute for Biomedical Engineering (OCIBME) since June 2016. He has also been an Adjunct Professor with the Department of Electrical and Computer Engineering, Royal Military College, Kingston, ON, since July 2015. He holds two patents and two disclosures of invention. He is the author of 160 journal articles and conference papers. He was involved in organizing several successful the IEEE conferences. His research interests include signal processing, biomedical signal processing, communication, and pattern classification.

Dr. Rajan served as a Board Member for the IEEE Canada from 2010 to 2018. He serves as the Chair for the IEEE Ottawa EMBS and the AESS Chapters. He received the IEEE MGA Achievement Award in 2012 and the IEEE contributions from the Queen Elizabeth II Diamond Jubilee Medal in 2012. His outstanding service recognized by the IEEE Canada through the 2016 W.S. Read Outstanding Service Award. He serves as a reviewer for the several IEEE journals and conferences.



**SIMON J. LEE** received the M.Phys. degree in physics from the University of Warwick, Coventry, U.K., in 2000, and the Ph.D. degree in physics from the University of Liverpool, U.K., in 2004. He held a postdoctoral position in magnetism, Liverpool, in 2007. He was with the Knowledge Transfer Partnership Project on advanced signal processing techniques in maritime radar as a collaboration of dB Research Ltd., and the University of Liverpool. He is currently a Principal Researcher in examining improvements in image processing and tracking in vessel traffic services systems with Denbridge Marine Ltd., Birkenhead, U.K.