Efficient Recurrent Neural Network for Classifying Target and Clutter: Feasibility Simulation of Its Real-Time Clutter Filter for a Weapon Location Radar

Il-Suek Koh^{1,*} · Hyun Kim² · Sang-Hyun Chun² · Min-Kil Chong²

Abstract

The classification of radar targets and clutter has been the subject of much research. Recently, artificial intelligence technology has been favored; its accuracy has been drastically improved by the incorporation of neural networks and deep learning techniques. In this paper, we consider a recurrent neural network that classifies targets and clutter sequentially measured by a weapon location radar. A raw dataset measured by a Kalman filter and an extended Kalman filter was used to train the network. The dataset elements are time, position, radial velocity, and radar cross section. To reduce the dimension of the input features, a data conversion scheme is proposed. A total of four input features were used to train the classifier and its accuracy was analyzed. To improve the accuracy of the trained network, a combined classifier is proposed, and its properties are examined. The feasibility of using the individual and combined classifiers as a real-time clutter filter is investigated.

Key Words: Clutter Filter, Radar Clutter Classification, Recurrent Neural Network.

I. INTRODUCTION

Recently, artificial intelligence (AI) technology has been successfully incorporated in a wide range of applications with emphasis on image and language processing. Due to the rapid development of computer technology, deep learning (DL) in an artificial neural network (ANN) has succeeded by using a massive dataset [1, 2]; this has drastically improved the ANN's precision. Many ANN structures, such as the convolutional neural network (CNN), recurrent neural network (RNN), and deep belief network (DBN), have been proposed, each having its advantages and disadvantages. A CNN can show performance comparable to that of an RNN for sequential data processing

but has a higher computational load [3]. Therefore, RNN networks have been commonly applied to sequential data training in language recognition and time-series analysis [4, 5].

In the radar industry, machine learning (ML) or DL techniques have been used in diverse areas such as synthetic aperture radar (SAR) applications [6, 7], jamming detection and classification [8], target detection and recognition [9–11], clutter suppression [12], and radar resource management [13]. In this paper, an ML technique is used to classify the artillery projectiles and radar clutter that are tracked by the weapon location radar (WLR) developed by LIG Nex1. This radar can simultaneously detect and track artillery projectiles fired by several guns. As a result, the radar data is sequentially measured in the time do-

Manuscript received March 3, 2021; Revised May 12, 2021; Accepted June 9, 2021. (ID No. 20210303-032J)

¹Department of Electronic Engineering, Inha University, Incheon, Korea.

²Radar R&D Center, LIG Nex1, Yongin, Korea.

*Corresponding Author: Il-Suek Koh (e-mail: ikoh@inha.ac.kr)

This is an Open-Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/4.0>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

© Copyright The Korean Institute of Electromagnetic Engineering and Science.

main. For this kind of classification, an RNN can be used to increase the radar's tracking capability when clutter tracking is terminated at the initial tracking process. It is important to reduce the probability of misclassifying the target as clutter. Also, the classifier design should be numerically efficient and provide a robust and highly reliable classification probability, even for an insufficient or highly imbalanced dataset, common in a radar measurement campaign. The WLR's target classification is reported in [14]; it only considers targets without clutter data.

In Section II, the measured dataset and its preparation for RNN training are addressed. A feature dimension reduction scheme is proposed. Then, in Section III, the performance of the trained RNN is assessed, and a simple method of enhancing its accuracy is proposed. Finally, the feasibility of using the RNN classifier to perform real-time rejection of clutter is investigated.

II. DATA PREPARATION

The WRL collected the dataset using two tracking filters, a simple Kalman filter (KF) and an extended Kalman filter (EKF). The filter designs were based on an interacting multiple model (IMM) that assumed constant velocity/acceleration of the measured objects. The EKF updated the target position in range and sine space angle (RUV coordinates): $U = \sin(Az)$ and $V = \sin(EI)$. Here, Az and EI are the azimuth and elevation angles in antenna coordinates where the antenna is in the xy-plane. The overall precision of the dataset collected by the EKF may be generally higher than that of the KF which can reduce the error caused by the nonlinearity of the coordinate transformation [15]. Four artillery projectiles were measured: gun heavy/light (GH/GL), mortar heavy/light (MH/ML), rocket heavy (RH), and rocket-assisted projectile (RAP). Table 1 shows the total number of targets and the clutter dataset. For the EKF datasets, only GH and MH have been measured so far.

Since the radar detected and tracked many clutters returns during the measurement, the clutter dataset is much larger than that of the other targets. The clutter is mainly categorized into two major object types: natural ones like clouds or birds and manmade ones like fixed- or rotary-wing aircraft. The raw measurement data consists of eight different data sets collected at discrete time points, t_n (in s) for $n = 1 \dots k$, among which three sets are used to train and test an ANN: target position $\vec{r}(t_n)$ (in m), radial velocity $v_r(t_n)$ (in m/s), and radar cross-

section (RCS) (in m^2) or signal-to-noise ratio (SNR). The position was measured in antenna coordinates and transformed based on the inertial navigation system to local east, north, up (ENU) Cartesian coordinates for the radar location.

1. Feature Selection and Transformation

Target tracking started when the target was detected and ended when sufficient tracking data was collected to extract ballistic parameters like the drag coefficient. During tracking, the data was updated at every sample time. When the radar temporarily lost the target track, it tried to detect the target at the next sample time and predict its position without an EK or EKF update (memory track). If the track was successfully restored, then the radar kept tracking the target. If not, the track was terminated after 1–5 trials [16]. This means that the data sequence length varies and can have missing values. Since the missing data may negatively affect the performance of the classifier, the data is linearly interpolated at the missing points [17]. All the initial times and positions of the data $t_n - t_1$ and $\vec{r}(t_n) - \vec{r}(t_1) = (E, N, U)$ in ENU coordinates.

Targets usually fly in a specific direction, but true clutter tracks are typically random. Thus, their most distinguishing feature might be their position sequence. Fig. 1 shows 10 samples of the adjusted position of the MH shell and clutter collected by the KF. Most of the clutter tracks are random, but some

Fig. 1. Measured position samples of KF dataset in ENU coordinates: (a) MH and (b) clutter.

Table 1. Number of measured datasets for four targets and clutter for the KF and EKF filters

	GH/GL	MH/ML	RH	RAP	Clutter	Total
KF	509	144	85	41	3,571	4,345
EKF	229	23	0	0	2,975	3,227

are very similar to that of the target, for example, "a" in Fig. 1(b). Since MH shells fly at a lower altitude than other targets, the received signal is highly contaminated by ground clutter as can be seen in Fig. 1(a). The RCSs of the artillery projectiles are very small, while that of the clutter is widely distributed as seen in Fig. 2. Thus, RCS may not be a suitable feature for this classification problem. Also, radial velocity does not clearly distinguish targets from clutter.

To maximally separate artillery projectile returns from clutter, positions were converted into three more features called Case 1, 2, 3, and 4. Case 1 is the measured position, (E, N, U) . Case 2 is the position difference as in $\Delta \vec{r}(t_n) = \vec{r}(t_{n+1}) - \vec{r}(t_n)$. To reduce the dimension of this feature, the normal vector to a plane containing $\vec{r}$ and $-\hat{U}$ is considered. Since the normal vector is two-dimensional, the slope of a line orthogonal to the line from the origin to (E, U) may be a good alternative; it is one-dimensional and is simply calculated by $-E/U$. This is called Case 3. For Case 4, a similar slope is computed based on the differences as in $-\Delta E(t_n)/\Delta N(t_n)$. Fig. 3 shows the probability density function (pdf) of four KF positions where the y-coordinate is plotted for Cases 1 and 2. Fig. 4 shows the identical pdfs for the EKF positions. In Figs. 3 and 4, for clear comparisons, the means of Case 2 and 4 are added by 2000 and 10, respectively. Similarly, the means of Case 2 and 4 are larger by 3000 and 10, respectively. For the training/test datasets, the position, radial velocity, and RCS were selected. Hence, the dimension of the input features is five for Cases 1 and 2; four for Cases 3 and 4. Of the whole dataset, 70% was randomly selected for training and 30% for tests.

III. TRAINING NETWORK AND PERFORMANCE ASSESSMENT

Binary and multiclass classifiers were trained by the weighted cross-entropy loss method to partially compensate for the imbalance of the dataset as seen in Table 1 [18]. The weight was simply calculated by the ratio of the size of target and clutter data to the total data size. The binary classifier sorts objects into two categories as "target" and "clutter," while the multiclass

Fig. 2. Probability density function of RCS collected by KF for four targets and clutter.

(a)

(b)

Fig. 3. Probability density function of positions collected by the KF: (a) the y-coordinate for Cases 1 and 2 and (b) the slope for Cases 3 and 4.

(a)

(b)

Fig. 4. Probability density function of positions collected by the EKF: (a) the y-coordinate for Cases 1 and 2 and (b) the slope for Cases 3 and 4.

classifier produces five categories of four types of artillery projectiles and clutter.

The ANN consists of two long short-term memory (LSTM) layers as regression and classification layers. The regression layer receives the input sequence, and the classification layer outputs a final classification label. The number of hidden cells for the first and second layers are denoted as N_1 and N_2 , respectively. To prevent the ANN from overfitting, a dropout layer is inserted between the two LSTM layers, and the dropout factor is assumed to be 0.1. The RNN is trained over 100 epochs. For fast training, the data is divided into a batch of data whose size is fixed at 50. At each epoch, the training data is randomly shuffled. First, the RNN is trained by the KF dataset since it is larger than the EKF dataset. Table 2 shows the accuracy of the binary RNN classifier for several values of N_1 and N_2 . Four different networks can be trained as classifier k using the Case k dataset for $k = 1, 2, 3$, and 4.

Table 3 shows the accuracy of the multiclass classifier. Due to

the reduction of one feature dimension, the accuracy of classifier 3(4) is slightly less than that of classifier 1(2), but it can be seen that accuracy for all classifiers is almost constant for two classifications.

Table 4 shows the accuracy of the two classifiers trained by the EKF dataset for a fixed $N_1 = N_2 = 20$. Although the number of datasets and the types of artillery projectiles are smaller than those for the KF dataset, the accuracy is improved. The multiclass classification probabilities are lower than those in [14] because the effect of clutter is much larger, but the EKF results are comparable to those in [14]. Based on Tables 2–4, it can be seen that 20 hidden cells for two layers may be sufficient. For the rest of the simulations, therefore, 20 cells were assumed for two LSTM layers and the binary classifier was further investigated.

To increase the size of the test data, the training and test datasets were combined into one test dataset. Table 5 shows the accuracy of two binary classifiers. Since 100 epochs are not sufficient for complete training, the overall accuracy of the classifiers in Table 4 degenerated as shown in Table 5, especially for the KF. The accuracy of an RNN classifier, however, is always higher than that of the best conventional tree algorithms such as the bagging tree algorithm.

When the dataset is sufficient, a single classifier trained by all input features may produce a good performance, but in this paper, the size of the dataset is not large. More hidden cells are required to increase the accuracy of the classifier which is not numerically efficient. Since the input features for each dataset case can capture a different physical mechanism of the track as seen in Figs. 3 and 4, a simple combination of two classifiers can significantly decrease misclassification probability. For example, classifier 1 and 3 could be used as a primary and a secondary classifier, respectively. If the two classifiers predict the same label,

Table 2. Accuracy (%) of the binary classifier trained by the KF dataset

		$N_2 = 10$	$N_2 = 20$	$N_2 = 30$	$N_2 = 40$
$N_1 = 10$	Case 1	95.4	95.6	95.5	95.4
	Case 2	95.3	94.3	95.6	96.2
	Case 3	93.0	95.3	95.3	95.4
	Case 4	94.8	95.8	95.5	96.0
$N_1 = 20$	Case 1	96.2	96.5	96.0	95.9
	Case 2	95.2	95.5	95.5	95.9
	Case 3	94.6	94.8	94.8	94.1
	Case 4	95.4	95.6	94.2	95.5
$N_1 = 30$	Case 1	96.4	96.6	96.1	95.7
	Case 2	95.3	94.8	95.2	94.6
	Case 3	94.3	94.8	95.5	94.5
	Case 4	96.2	96.4	96.0	95.6

Table 3. Accuracy (%) of the multiclass classifier trained by the KF dataset

		$N_2 = 10$	$N_2 = 20$	$N_2 = 30$	$N_2 = 40$
$N_1 = 10$	Case 1	93.6	92.8	94.2	94.0
	Case 2	93.2	94.8	94.0	94.7
	Case 3	90.4	92.1	91.0	91.1
	Case 4	91.7	93.4	92.6	94.1
$N_1 = 20$	Case 1	93.0	94.1	94.3	94.3
	Case 2	95.0	95.1	95.4	95.1
	Case 3	92.5	91.4	91.4	93.1
	Case 4	91.5	92.7	93.6	92.5
$N_1 = 30$	Case 1	94.9	94.0	93.9	95.5
	Case 2	94.2	94.4	95.7	96.0
	Case 3	91.7	92.3	92.9	93.0
	Case 4	92.0	93.0	93.5	92.7

Table 4. Accuracy (%) of classifiers trained by the EKF dataset

	Case 1	Case 2	Case 3	Case 4
Binary	98.4	99.6	96.3	96.6
Multiclass				
Cutter	99.0	99.4	99.9	98.5
GH/GL	98.7	96.8	96.9	93.5
MH/ML	100	100	100	100

Table 5. Accuracy (%) of the binary classifier for the combined dataset

	Case 1	Case 2	Case 3	Case 4
KF	90.8	84.2	91.1	89.3
EKF	96.8	97.4	86.5	87.8

it becomes the final label of the combined classifier. If not, the label produced by the primary classifier flips to the other label. Fig. 5 shows the block diagram of the combined classifier; it can be extended to a real-time classifier with a given threshold probability. The number of erroneous predictions is summarized in Tables 6 and 7 for two datasets. For this simulation, the threshold is assumed as 1.

The diagonal numbers in Tables 6 and 7 are the result of a sole classifier, while the off-diagonal numbers are those for the combined classifier. For example, 181 (5.1%) at the second row and third column in Table 6 is the mislabel number (probability) if classifier 1 and classifier 2 are used as the primary and secondary classifiers, respectively. The combination of classifier 2 with 3 or 4 can provide the highest accuracy. For example, the combination of classifier 2 and 3 can reduce the misclassification probability from 3.6% to 1.0% and 18.4% to 2.7% for “target as clutter” and “clutter as target” labels, respectively, compared with those of the sole classifier 2 for the KF dataset. Table 7 shows a

Fig. 5. The block diagram of combined classifier with threshold.

Table 6. Number of classifier labeling errors for the KF dataset

		Case 1	Case 2	Case 3	Case 4
Clutter as target (n = 3,571)	Case 1	339 (9.5)	181 (5.1)	97 (2.7)	120 (3.4)
	Case 2	-	657 (18.4)	96 (2.7)	128 (3.6)
	Case 3	-	-	248 (7.0)	119 (3.3)
	Case 4	-	-	-	325 (9.1)
Target as cluster (n = 774)	Case 1	62 (8.0)	27 (3.5)	39 (5.0)	40 (5.2)
	Case 2	-	28 (3.6)	8 (1.0)	7 (0.9)
	Case 3	-	-	140 (18.1)	110 (14.2)
	Case 4	-	-	-	140 (18.1)

Values are presented as number (%).

Table 7. Number of classifiers labeling errors for the EKF dataset

		Case 1	Case 2	Case 3	Case 4
Clutter as target (n = 2,975)	Case 1	95 (3.2)	23 (0.8)	11 (0.4)	2 (0.1)
	Case 2	-	83 (2.8)	35 (1.2)	29 (1.0)
	Case 3	-	-	435 (14.6)	211 (7.1)
	Case 4	-	-	-	392 (13.2)
Target as cluster (n = 252)	Case 1	8 (3.2)	0 (0)	0 (0)	0 (0)
	Case 2	-	0 (0)	0 (0)	0 (0)
	Case 3	-	-	2 (0.8)	1 (0.4)
	Case 4	-	-	-	2 (0.8)

Values are presented as number (%).

similar reduction of mislabeling probability for the EKF dataset. Hence, a combination of classifier 2 with 3 or with 4 may be appropriate for both the KF and EKF datasets. Fig. 6 shows the target and clutter tracks mislabeled as clutter and target by the combined classifier 1(3) and 4 for the EKF dataset. The mislabeled target and clutter tracks are similar to those in Fig. 1(b) and 1(a), respectively.

Since the trained RNN is swallow, its real-time computation may be feasible. To estimate the real-time performance of the

Fig. 6. The misclassified tracks for the EKF dataset in ENU coordinates: (a) target using classifier 3 and 4, (b) clutter using classifier 1 and 4.

classifier, the features were sequentially inputted to the network. If the predicted probability of one of two labels was greater than a given threshold, then the network would output the classification label. Table 8 shows the overall accuracy of the real-time simulation of the four trained classifiers for two threshold probabilities, 80% and 90%. The mean ratio of the required length of the sequence to the total length for termination is also summarized. About 50% of the sequence is required for termination. Tables 9 and 10 show the misclassification number (probability) for the proposed combined classifier for the KF and EKF datasets, respectively. The combination of classifiers can drastically improve the final accuracy.

Similar to the previous analysis, the combined classifiers 2 with 3 and 1 with 3 may be the best for the KF and EKF datasets, respectively. After a 58% tracking process, the mislabel probability for "target as clutter" was reduced from 4.5% to 1.6%

for the KF dataset and from 4.5% to 2.4% for the EKF dataset with the 90% threshold probability, respectively.

The computer used to train the proposed network comprised an Intel Core i7-4930K CPU running at 3.40 GHz with 64 GB of RAM. The training time was within 5–10 minutes.

IV. CONCLUSION

To classify artillery projectiles and clutter tracked by a weapon location radar, an RNN classifier was trained with a real measured dataset. The data was collected by the radar using two tracking filters, a KF and an EKF. The sizes of the KF and EKF datasets are 4,345 and 3,227, respectively, and are highly imbalanced. The dataset clutter portion is around 82% and 92% for the KF and EKF, respectively. KF data for four types of artillery projectiles and EKF data for two of those types were collected. To

Table 8. Accuracy (%) of the real-time classifier and mean ratio of the required length to the total length of the sequence for termination

	Threshold	Accuracy				Mean of required sequence length/total length			
		Case 1	Case 2	Case 3	Case 4	Case 1	Case 2	Case 3	Case 4
KF	80%	88.2	82.0	88.0	88.3	31.8	40.2	35.2	41.3
	90%	89.3	83.1	90.3	89.3	44.4	52.8	48.7	57.3
EKF	80%	90.6	93.7	86.1	86.5	22.8	23.9	44.6	40.7
	90%	90.7	95.3	85.7	86.7	30.5	38.9	58.3	52.0

Table 9. Number of mislabels for real-time classifiers for the KF set

			Case 1	Case 2	Case 3	Case 4
Threshold = 80%	Clutter as target (n = 3,571)	Case 1	324 (9.1)	195 (5.5)	56 (1.6)	106 (3.0)
		Case 2	-	736 (20.6)	73 (2.0)	115 (3.2)
		Case 3	-	-	194 (5.4)	73 (2.0)
		Case 4	-	-	-	257 (7.2)
	Target as clutter (n = 774)	Case 1	191 (24.7)	39 (5.0)	145 (18.7)	147 (19.0)
		Case 2	-	47 (6.1)	39 (5.0)	24 (3.1)
		Case 3	-	-	329 (42.5)	176 (22.7)
		Case 4	-	-	-	251 (32.4)
Threshold = 90%	Clutter as target (n = 3,571)	Case 1	325 (9.1)	191 (5.4)	84 (2.4)	112 (3.1)
		Case 2	-	698 (19.6)	94 (2.6)	119 (3.3)
		Case 3	-	-	235 (6.6)	102 (2.9)
		Case 4	-	-	-	286 (8.0)
	Target as clutter (n = 774)	Case 1	142 (18.3)	29 (3.8)	98 (12.7)	109 (14.1)
		Case 2	-	35 (4.5)	12 (1.6)	13 (1.7)
		Case 3	-	-	188 (24.3)	132 (17.1)
		Case 4	-	-	-	178 (23.0)

Values are presented as number (%).

Table 10. Number of mislabels for real-time classifiers for the EKF set

			Case 1	Case 2	Case 3	Case 4
Threshold = 80%	Clutter as target (n = 2,975)	Case 1	61 (2.1)	1 (0)	3 (0.1)	0 (0)
		Case 2	-	51 (1.7)	19 (0.6)	22 (0.7)
		Case 3	-	-	420 (14.1)	205 (6.9)
		Case 4	-	-	-	404 (13.6)
	Target as clutter (n = 252)	Case 1	242 (96.0)	142 (56.3)	18 (7.1)	21 (8.3)
		Case 2	-	151 (59.9)	19 (7.5)	23 (9.1)
		Case 3	-	-	28 (11.1)	22 (8.7)
		Case 4	-	-	-	31 (12.3)
Threshold = 90%	Clutter as target (n = 2,975)	Case 1	60 (2.0)	12 (0.4)	2 (0.1)	0 (0)
		Case 2	-	74 (2.5)	31 (1.0)	28 (0.9)
		Case 3	-	-	445 (15.0)	225 (7.6)
		Case 4	-	-	-	406 (13.7)
	Target as clutter (n = 252)	Case 1	239 (94.8)	69 (27.4)	6 (2.4)	10 (4.0)
		Case 2	-	77 (30.6)	8 (3.2)	9 (3.6)
		Case 3	-	-	17 (6.8)	12 (4.8)
		Case 4	-	-	-	22 (8.7)

Values are presented as number (%).

clarify the radar signature difference between the artillery projectiles and clutter, the input data were selected and converted to four different input feature expressions. Then, binary and multiclass RNN classifiers were trained with the weighted cross-entropy loss method to compensate for the class imbalance. It was shown that the performance of the classifiers is insensitive to the number of hidden cells. Thus, 20 hidden cells were assumed for two layers for further analysis. The multiclass classifier's accuracy is less than that of the binary case, especially for the KF data. To increase the accuracy of the binary classifier, two classifiers were combined. When classifiers 2 and 3 are combined against the KF dataset, false probabilities decrease from 3.6% to 1.0% and from 18.4% to 2.7% for the mislabeling of "target as clutter" and "clutter as target," respectively. A similar reduction can be observed for the EKF dataset. For real-time applications, the trained RNN can be terminated when the predicted probability of one of two labels is greater than a given threshold. Below about 60% of the tracking process, usually 50%, the network can terminate. It is recommended that classifier 3 be combined with classifier 2 for the KF dataset and with classifier 1 for the EKF dataset. For these combinations, the mislabel of "target as clutter" probability is around 1.6% and 2.4% for the KF and EKF datasets, respectively. When more EKF target data is available, the final accuracy can be improved.

REFERENCES

- [1] A. Shrestha and A. Mahmood, "Review of deep learning algorithms and architectures," *IEEE Access*, vol. 7, pp. 53040-53065, 2019.
- [2] M. Z. Alom, T. M. Taha, C. Yakopcic, S. Westberg, P. Sidiqi, M. S. Nasrin, et al., "A state-of-the-art survey on deep learning theory and architectures," *Electronics*, vol. 8, no. 3, article no. 292, 2019. <https://doi.org/10.3390/electronics8030292>
- [3] M. Kudo, J. Toyama, and M. Shimbo, "Multidimensional curve classification using passing-through regions," *Pattern Recognition Letters*, vol. 20, no. 11-13, pp. 1103-1111, 1999.
- [4] M. Husken and P. Stagge, "Recurrent neural networks for time series classification," *Neurocomputing*, vol. 50, pp. 223-235, 2003.
- [5] S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," 2018 [Online]. Available: <https://arxiv.org/abs/1803.01271>.
- [6] X. X. Zhu, D. Tuia, L. Mou, G. S. Xia, L. Zhang, F. Xu, and F. Fraundorfer, "Deep learning in remote sensing: a comprehensive review and list of resources," *IEEE Geoscience and Remote Sensing Magazine*, vol. 5, no. 4, pp. 8-36, 2017.
- [7] D. Hong, L. Gao, N. Yokoya, J. Yao, J. Chanussot, Q. Du, and B. Zhang, "More diverse means better: multimodal deep learning meets remote-sensing imagery classification," *IEEE*

This work is supported by LIG Nex1 research fund.

- Transactions on Geoscience and Remote Sensing*, vol. 59, no. 5, pp. 4340-4354, 2020.
- [8] K. Grover, A. Lim, and Q. Yang, "Jamming and anti-jamming techniques in wireless networks: a survey," *International Journal of Ad Hoc and Ubiquitous Computing*, vol. 17, no. 4, pp. 197-215, 2014.
- [9] E. V. Carrera, F. Lara, M. Ortiz, A. Tinoco, and R. Leon, "Target detection using radar processors based on machine learning," in *Proceedings of 2020 IEEE ANDESCON*, Quito, Ecuador, 2020, pp. 1-5.
- [10] A. P. Sligar, "Machine learning-based radar perception for autonomous vehicles using full physics simulation," *IEEE Access*, vol. 8, pp. 51470-51476, 2020.
- [11] M. Akcakaya, S. Sen, and A. Nehorai, "A novel data-driven learning method for radar target detection in non-stationary environments," *IEEE Signal Processing Letters*, vol. 23, no. 5, pp. 762-766, 2016.
- [12] D. Callaghan, J. Burger, and A. K. Mishra, "A machine learning approach to radar sea clutter suppression," in *Proceedings of 2017 IEEE Radar Conference (RadarConf)*, Seattle, WA, 2017, pp. 1222-1227.
- [13] M. Shaghghi and R. S. Adve, "Machine learning based cognitive radar resource management," in *Proceedings of 2018 IEEE Radar Conference (RadarConf18)*, Oklahoma City, OK, 2018, pp. 1433-1438.
- [14] S. J. Park and H. S. Jin, "Classification type of weapon using artificial intelligence for counter-battery radar paper title," *Journal of IKEEE*, vol. 24, no. 4, pp. 921-930, 2020.
- [15] X. Tian and Y. Bar-Shalom, "Coordinate conversion and tracking for very long range radars," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 45, no. 3, pp. 1073-1088, 2009.
- [16] J. W. Rim, S. Oh, and I. S. Koh, "High resolution radar model to simulate detection/tracking performance of multi-function radar in war game simulator," *The Journal of Korean Institute of Electromagnetic Engineering and Science*, vol. 30, no. 1, pp. 70-78, 2019.
- [17] Z. C. Lipton, D. C. Kale, C. Elkan, and R. Wetzell, "Learning to diagnose with LSTM recurrent neural networks," 2017 [Online]. Available: <https://arxiv.org/abs/1511.03677>.
- [18] Y. Ho and S. Wookey, "The real-world-weight cross-entropy loss function: modeling the costs of mislabeling," *IEEE Access*, vol. 8, pp. 4806-4813, 2020.

Il-Suek Koh

received B.S. and M.S. degrees in electronic engineering from Yonsei University, Seoul, Korea in 1992 and 1994, respectively, and he obtained his Ph.D. degree from the University of Michigan, Ann Arbor, MI, USA in 2002. In 1994, he joined LG Electronics Ltd., Seoul, as a Research Engineer. Currently, he is a professor at Inha University, Incheon, Korea. His research interests include numerical

and analytical methods for electromagnetic fields and radar applications.

Sang-Hyun Chun

received B.S., M.S., and Ph.D. degrees in microwave engineering from Kwangwoon University, Seoul, Korea, in 2005, 2007, and 2011, respectively. He is currently a research engineer in LIG Nex1, Korea. His main interests are in the area of transmit/receive modules and radar systems.

Hyun Kim

received B.S., M.S., and Ph.D. degrees in electrical and electronics engineering from Yonsei University, Seoul, Korea, in 2002, 2005, and 2011, respectively. He is currently a research engineer in LIG Nex1, Korea. His main interests are in the area of numerical analysis, phased array antenna, and radar systems.

Min-Kil Chong

received B.S. and M.S. degrees in electronics engineering from Gyeongsang National University, Korea, in 1994 and 1996, respectively. He is currently a research engineer in LIG Nex1, Korea. His main interests are in the area of transmit-receive module, active phased array antenna, and radar systems.