

Improved Shape Parameter Estimation in K Clutter with Neural Networks and Deep Learning

José Raúl Machado Fernández, Jesús de la Concepción Bacallao Vidal

Instituto Superior Politécnico José Antonio Echeverría, Cuba

Abstract — The discrimination of the clutter interfering signal is a current problem in modern radars' design, especially in coastal or offshore environments where the histogram of the background signal often displays heavy tails. The statistical characterization of this signal is very important for the cancellation of sea clutter, whose behavior obeys a K distribution according to the commonly accepted criterion. By using neural networks, the authors propose a new method for estimating the K shape parameter, demonstrating its superiority over the classic alternative based on the Method of Moments. Whereas both solutions have a similar performance when the entire range of possible values of the shape parameter is evaluated, the neuronal alternative achieves a much more accurate estimation for the lower Fig.s of the parameter. This is exactly the desired behavior because the best estimate occurs for the most aggressive states of sea clutter. The final design, reached by processing three different sets of computer generated K samples, used a total of nine neural networks whose contribution is synthesized in the final estimate, thus the solution can be interpreted as a deep learning approximation. The results are to be applied in the improvement of radar detectors, particularly for maintaining the operational false alarm probability close to the one conceived in the design.

Keywords — Sea Clutter, K Distribution, Shape Parameter Estimation, Artificial Neural Networks, Deep Learning

I. INTRODUCTION

RADARS are responsible for detecting nearby targets using electromagnetic waves [1]. The echo amplitude received by the radar, besides carrying the target's information, contains a distortion introduced by surrounding objects. In the particular case of the operation in sea environments, the distortion signal is originated in the sea surface and it's therefore called sea clutter [2].

Among the various ways of improving radars' performance, clutter discrimination is frequently addressed in the literature [3]. Although other models can be used [4-6], the K distribution is widely recognized as the best choice for sea clutter representation [7]. Consequently, several solutions have been proposed taking as a priori condition the assumption of K clutter [8-10].

Radar classic detectors, such as the CA-CFAR (Cell Averaging - Constant False Alarm Rate), use, as a general rule, a fixed adjustment factor to define a desired value of false alarms probability [11]. However, when a variation appears in the clutter's statistics, a correction of the adjustment factor is necessary to ensure the operating false alarm probability remains close to the one conceived in the design [12].

The statistical variation of the background signal translates mathematically into a modification of the shape parameter of the assumed model. The problem can be then reduced to a search of the

shape parameter using a finite set of samples.

The more popular technique for estimating the K shape parameter is the Method of Moments (MoM) which uses the 2nd and 4th order algebraic moments [13]. The MoM is also used in the estimation under Weibull clutter, through the computing of the first two moments. A previous research related to the Weibull model has shown that Artificial Neural Networks (ANN) are able to replace the MoM in some cases, achieving a more accurate estimate without incurring into a high computational cost [14, 15].

In view of the presented analysis, the authors established as the goal of the investigation the design and training of a neural network capable of estimating the K shape parameter from a finite set of samples. Initially they expected to find the proper solution by modifying multiple internal variables or conditions of the neural network and using groups of 3000 samples, according to what was done in [15]. However, this approach was insufficient to achieve good results.

Then, considering the recommendations given in [16] and adopting an strategy similar to [17], they decided to divide the region of possible values of the shape parameter in five reduced estimation intervals. The new solution, which includes nine neural networks, achieved significantly better results than those exhibited by the MoM and constitutes, therefore, the contribution of this paper. The design can be interpreted as a deep learning approach to the K parameter shape estimation given the high degree of integration of the nine networks.

II. MATERIALS AND METHODS

This paper's results were achieved by performing simulations in MATLAB 2011 software. The current section is devoted to facilitate the replication of the experiment by third parties. Therefore, it describes (A) the way in which clutter samples were generated and (B) the methods considered for obtaining K shape parameter estimates.

A. Samples used in the Simulations

The following PDF was employed for the K distribution [18]:

$$f_x(x) = \frac{4c}{\Gamma(v)} (cx)^v K_{v-1}(2cx) \quad (1)$$

Where c is the scale parameter, v is the shape parameter, $\Gamma(\cdot)$ is the Gamma function and $K(\cdot)$ is the Bessel function of second kind and order $v - 1$. A complete definition of the functions related to the K distribution can be found in [19]. Actually, the MATLAB functions implemented and validated in [19] were used in this study.

As a design principle, it was decided that the neural network should use histograms as a distinctive feature of the data. Replicating what was done in [15], a quantity of 3000 samples was selected to assemble each histogram.

By observing multiple sets of histograms, the authors concluded

that the greatest amount of information was concentrated below the amplitude equal to five. Consequently, samples with magnitude greater than five were ignored when histograms were prepared.

Three sets of samples, in which the shape parameter (ν) was varied from 0.1 to 30 according to the investigations found in [20-22], were created. Moreover, the scale parameter (c) was maintained as $c = \sqrt{\nu}$ to fix the average of the samples to one [18].

The generated data sets have the structure shown in Fig. 1. Each of the sets covered the range of values of ν between 0.1 and 30 in different ways.

Set A started with a group of 3000 samples corresponding to $\nu = 0.1$. The next ten groups maintained the same ν value, representing different outcomes of the same configuration of parameters. Then, group number 11 resulted from adding 0,01 to the shape parameter before generating the samples. Again, the next ten groups maintained the parameter configuration. Next, the mechanism was repeated until 30 000 samples sets were created, with a change in ν every ten groups. The total number of samples of Set A was 90 million.

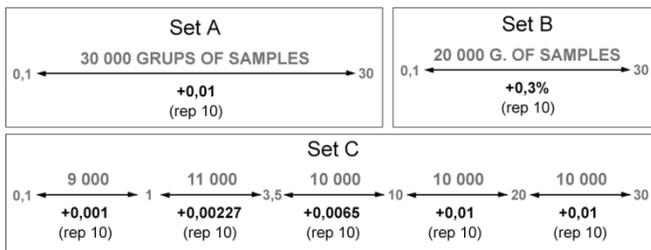


Fig. 1. Data Sets Prepared for the Training and test of Neural Networks in the Estimation of the K Shape Parameter.

Set B and Set C also changed the ν value every ten groups but they followed different rules. Set B increased ν a 0,3% every ten groups, ensuring thus that the lower ν Fig.s were covered with a smaller step. According to recommendations given in [16], the high magnitudes of the shape parameter have a reduced influence in the heavy property of the K distribution tails. Therefore, there is no great interest in providing a reliable estimate for such values. In fact, the purpose of the Set B is to indicate the neural network to pay less attention to the estimation of high ν magnitudes.

Set C achieves a similar effect by applying different variation steps into five incremental length intervals. For the interval $0.1 < \nu < 1$ a variation step of 0,001 was used; for $1 < \nu < 3,5$ the step was of 0,00227; for $3,5 < \nu < 10$ of 0,0065; and for $10 < \nu < 20$ and $20 < \nu < 30$ of 0,01. Finally, Set C included 150 million samples distributed in 50000 groups of 3000 samples each, and was the one more intensively used in the neuronal training.

Sets A, B and C were created in that order in response to observed outcomes in the performed trials. The objective was to positively influence the performance of the neuronal solution. The reasons for the creation of each set are explained in the section “Results and Discussion”.

B. Methods for Obtaining K Shape Parameter Estimates

Several methods for estimating the K distribution parameters have been proposed in the literature [23-30]. In [31], a comparison of four different methods was performed, concluding that the MoM based on second and fourth moments [32] displays the best results. Additionally, the same method was used in [13] for obtaining estimates from samples taken at the coast of Taiwan.

For the above reasons, the authors selected the MoM that uses the second and fourth moments like the classical estimator of K parameters. According to this method, expression (2) is to be applied:

$$\hat{\nu} \approx \frac{1}{\left(\frac{m_4}{2m_2^2} - 1\right)} \quad \hat{c} \approx 2\sqrt{\frac{\nu}{m_2}} \quad (2)$$

Therefore, the objective of this research is to build a neural network based scheme that will exceed the performance of the previously described MoM.

For the design of the neural scheme, the authors took as a start point the solutions given in [15, 33-37] for different situations. Consequently, the initial configuration of the neural network internal variables was arranged according to Table I.

The first trained neural network was a classic three layers Perceptron. It had 50 inputs for reading 50 values resulting of samplings made on the histograms obtained from each 3000 samples group extracted from sets A, B and C. This network failed to outperform the MoM; so the authors, after trying several alternatives, decided to use a nine neural networks scheme, in which each net would read the same data and reach to an independent conclusion. The synthesis of these conclusions would produce the final estimate.

TABLE I
CONFIGURATION OF THE INTERNAL NEURAL NETWORK
VARIABLES

Network Variables	Choice
Network Type	Feed Forward Network (Multilayer Perceptron)
Training Function	BackPropagation (Levenberg-Marquardt)
Number of Layers	3 (Input Layer – Hidden Layer – Output Layer)
Transference or Activation Function	Hyperbolic Tangent Sigmoid (Hidden Layer), Lineal Transference Function (Output Layer)
Activation Order	Topological (Asynchronous Activation)
Error Measurement	Mean Square Error
Training Set Division	70% Training - 15% Validation - 15% Test
Samples' Presentation	Batch Training

The logical process that led to the design of the neural networks based solution is described in the next section. Five of the nine networks use a configuration very similar to the one presented in Table I; whereas the other four change the transfer function of the output layer to *Hyperbolic Tangent Sigmoid*. As a result, their output turns to be restricted to the range between 0 and 1. For a detailed understanding of the meaning of each of the parameters in Table I, the reader is referred to specialized literature [38].

III. RESULTS AND DISCUSSION

The current section shows evidence of the new neuronal solution's proper functioning and demonstrates its superiority after performing a comparison with the MoM. First, efforts are presented regarding the training of a single network for obtaining K shape parameter estimates. Then, the final scheme, obtained after trying different alternatives, is described. This scheme is based on the interaction of nine neural networks trained in an independent way to solve fractions of the problem at hand.

A. Design of a Single Network

Multiple trainings were executed with a neural network that used the configuration shown in Table I and processed the data from Set A. The size of the hidden layer was changed between 5 and 50 neurons,

without obtaining a significant improvement. Finally, it was found that the biggest success rate was achieved by using 30 neurons.

Aiming to improve the network’s performance, the authors made several modifications on the training process. First, they proceeded to omit some of the 50 inputs. So, they repeated the essays omitting the last 3, 5, 10, 15, 20 and 25 values from the tail of the input histograms. The results showed a reduction in the effectiveness. Then, values from the beginning of the input histograms were skipped from the training procedure. The experiments showed that the first 20 inputs could be omitted without significantly affecting the estimator’s performance. However, almost no improvement was achieved.

Two main problems prevented the performance’s improvement. The first one was the rapid convergence of the training. The process often concluded between iterations 15 and 20 because the network began to lose generalization and the validation made with a percentage of data stopped the training. The reader should note that this effect is common in *Early Stopping* based trainings. The authors chose a high limit on validation failures but the performance did not improve. It was noted that once the network begins to particularize its performance, it fails to generalize the behavior even after 30 iterations.

Searching for a slower convergence, *Scaled Conjugate Gradient* and *Bayesian Regularization* training algorithms were tested. They both succeeded in offering a slower convergence, and there was a slightly enhancement in the results. It was also decided to create a training set with more samples to see if a wider variety of presentation helped the network to generalize the results. To this end, trainings were repeated with a Set very similar to Set A but having 20 repetitions of ν instead of ten. Again, the results improved slightly.

The second problem was the poor performance in the $0.1 < \nu < 1$ region. Recommendations given in [16] clearly indicate the need for a very accurate estimate in this area, which is responsible for the heavy property of the K histograms tails. Heavy tails have a marked influence on the selection of the adjustment factor in detectors such as the CA-CFAR, and therefore, in the performance of radar detection schemes.

Accordingly, the authors created Set B, which includes more groups of samples in the region of reduced ν magnitudes. Trainings conducted with this group achieved a greater precision in the $0.1 < \nu < 1$ interval, maintaining the previously achieved overall performance.

Set C was also created as a way to expand the number of samples and place a larger amount of them in the regions of interest. Trainings conducted with this group positively contributed to improve the network performance.

Once the above described modifications were considered, together with others that were omitted to simplify the explanation, the authors arrived to the conclusion that follows. After using a set of 150 million samples (Set C), varying the size of the hidden layer from 5 to 50 neurons, altering training variables and placing more groups of samples in the region of interest, it was found that a single neural network is unable to outperform the MoM in the estimation of the K distribution shape parameter. Fig. 2 shows a detailed comparison of both estimators by displaying error histograms.

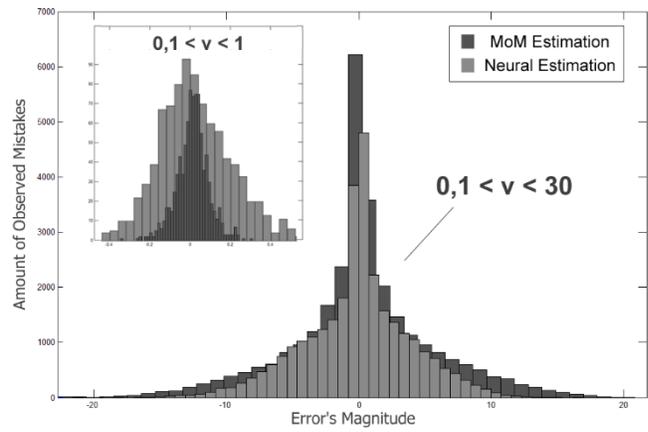


Fig. 2. Histograms of Errors Committed by the Neural and MoM Estimators.

If Fig. 2 is examined, it can be seen that even though the overall performance of the neural network is better than the MoM, this cannot be considered an advantage because the improvement is detrimental of the quality of the estimates in the region of interest ($0.1 < \nu < 1$). Inevitably, the neural network’s learning mechanism compels it to maintain a uniform performance through the entire search region. Such behavior is desirable in many situations but causes unwanted results in the estimation of the K shape parameter. In conclusion, it can be stated with certainty that the variations applied in [15] for training a single network to find the Weibull shape parameter, are not sufficient to achieved the same for the K case.

A. Two Stage Neural Solution

Given the difficulties encountered training a single network, the authors resolved to assess the estimation problem using multiple neural networks that would provide fractions of the solution. A similar principle was followed in [17] leading to suitable results.

However, unlike [17], two stages were used in the design, each one including several neural networks; being the response of the second stage determined by the output of the first one. The first stage was responsible for dividing the region into smaller sub-regions. Then, the second stage was in charge of producing an accurate estimate within each sub-region. Thus, the problem was divided in easier to process portions which would present a simplified problem to specialized networks.

Networks from the first stage were designed using the *Hyperbolic Tangent Sigmoid* function in the output layer. Note that this configuration differs from the one presented in Table I. The intention was to force the output values to be 0 or 1. So, each network from the first stage had a single output whose task was to decide which of two ν regions contained a given group of samples. Therefore, each network divided the search region in a portion that it classified as 0 and another that it classified as 1. This solution can be interpreted as form of pattern recognition.

In contrast, internal variables from second stage networks remained

TABLE II
ERROR INTERVAL FOR EACH TESTED BOUNDARY VALUE.

X	20	18	16	15	14	13	12	11	10	9	8
$X_2 - X_1$	22,12	22,46	23,36	24,03	23,89	24,88	24,52	24,18	24,42	18,98	18,80
X	7	6	5	4	3	2	1	0,95	0,90	0,85	0,80
$X_2 - X_1$	10,94	8,7	5,07	3,48	1,72	0,91	0,22	0,24	0,29	0,16	0,20

very similar to what was shown in Table I, and only suffered minor changes according to the comments made on section 3.1. These networks had no restrictions on the output values.

Next, the following section will explain how the first stage networks were designed. Understanding the results obtained at the output of the first stage is a key element in the justification of the search intervals intersection conceived for stage two nets.

B. B.1. First Stage Networks

The first stage of the neural solution sustains a complexity that has not been yet discussed. In order to divide two regions, it's necessary to place a boundary between them. Given the variability of the histograms that were assembled with a finite amount of samples, the continuity of the search region, and complexities associated with the training of artificial intelligence systems, it is virtually impossible to establish an absolute border without incurring in a certain degree of error. Consequently, the expected behavior after establishing a boundary is the one shown in Fig. 3.

After conducting a training for establishing a border (X) between Regions 0 and 1, an area of uncertainty (between X_1 and X_2) will always appear. This area will contain network mistakes on the election of the proper region. In the interval between X and X_1 , the network will make mistakes classifying groups of samples belonging to the Region 0 as Region 1 (errors 1/0). The opposite can be expected between X and X_2 , where groups of samples belonging to Region 1 will be classified as Region 0 (errors 0/1). This behavior is impossible to avoid, and the designer can only seek to reduce the interval between X_1 and X_2 as much as possible.

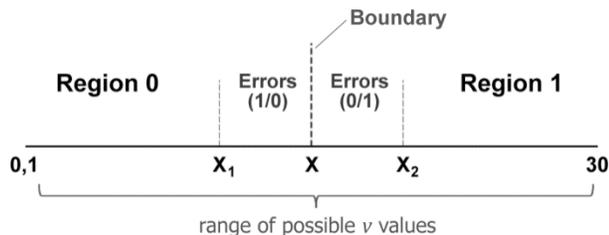


Fig. 3. Expected Neural Behavior after Conducting a Training for Dividing the Search Region in two sub-Regions.

Table II shows a measurement of the length of the error interval ($|X_2 - X_1|$) incurred by a neural network in relation to the selected X . Note that X may vary through the whole range of possible v values (from 0.1 to 30). The experiments correspond to the best network of ten neurons in the hidden layer after executing 20 trainings for each X .

Even though values in Table II are only an approximation, they allow appreciating the effect of X variation on the magnitude of the error. As it can be seen, as X gets reduced, the interval of mistakes gets smaller. The first value that is worth considering is $X = 7$ which provides an error of 10,94. Any value above $X = 7$ is completely unacceptable.

For the design of the first network for dividing the search region into two smaller sub-regions, the authors chose $X = 0.90$ and executed multiple trainings to find an improved error interval. A neural network, capable of operating with $X_1 = 0.8040$ and $X_2 = 0.983$, and having only five neurons in the hidden layer, was found. The results were achieved using the Set C, and were subsequently validated with Set A.

This first network, called *net 1*, it's able to clearly distinguish between the $0.1 < v < 0.8$ and $0.8 < v < 30$ regions, but it makes some mistakes in the $0.8 < v < 1$ portion. If *net 1* is placed in the

stage one, the next stage must have at least two networks operating with an interception on its operating regions: the first must operate in $0.1 < v < 1$ and the second in $0.8 < v < 30$. Thus, they will both perform the estimation in the $0.8 < v < 1$ interval, which is where the first stage makes mistakes.

Three additional networks for the first stage were later designed. Following a similar logic, each one divided the search region into two smaller sub-regions. So, *net 2* attempted to place a boundary close to $X = 1,5$, *net 3* close to 2,7 and *net 4* close to 5,7. The reader should note that, as it was pointed out in [16], the influence of the K shape parameter on the heavy property of the distribution becomes rapidly saturated with v increase. This means that the difference will be much greater between histograms with $v = 1$ and $v = 2$ than between histograms with $v = 11$ and $v = 12$.

C. B.2. Connecting both Stages

Fig. 4 summarizes the interconnection structure of the nine neural networks that were finally included the scheme. The four networks from the first stage make decisions on the region in which the data is located and choose which network from the second stage should be used to acquire the final result.

For example, *net 1* decides whether to use *net 1E* to get the result. If it concludes that it's not appropriate to use *net 1E*, the decision passes to *net 2*. Then, *net 2* decides if the data belong to the region for which *net 2E* was trained. Otherwise, the decision is passed to *net 3* which performs a similar operation. The *net 4* is the last one from the first stage and makes the final decision by choosing between *net 4E* and *net 5E*. Note that in many cases it won't be necessary to use *net 4*.

The error interval of each network from the first stage and the training region for each network from the second stage are revealed in Fig. 5. All networks use five neurons in the hidden layer, except *net 5E* which uses 20.

The network *net 1* seeks to find groups of samples with a v lower than 0,9 and send them to *net 1E*. It has an error interval of $0.8 < v < 1$. Therefore, the network *net 1E* must be able to process samples up to $v = 1$ and *net 2E* must be able to perform the estimation from $v = 0.8$.

The network *net 2* searches for groups of samples with $v < 1,5$ for sending them to *net 2E*. It has an error interval of $1,3 < v < 1,73$. Therefore, *net 2E* must be able to process samples up to $v = 1,73$ and *net 3E* must be able to perform the estimation from $v = 1,3$.

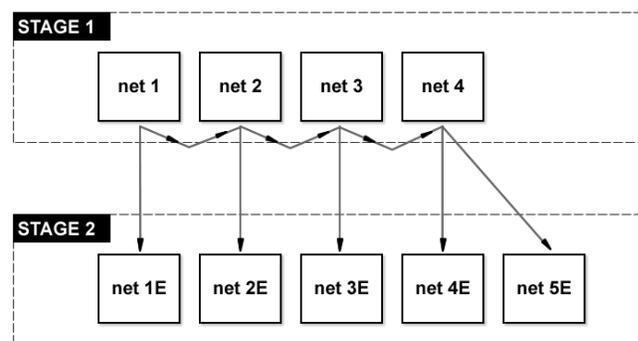


Fig. 4. Interconnection of Internal Networks from both Stages.

The network *net 3* searches for groups of samples with $v < 2,7$ for sending them to *net 3E*. It has an error interval of $1,92 < v < 3,6$. Therefore, *net 3E* must be able to process samples up to $v = 3,6$

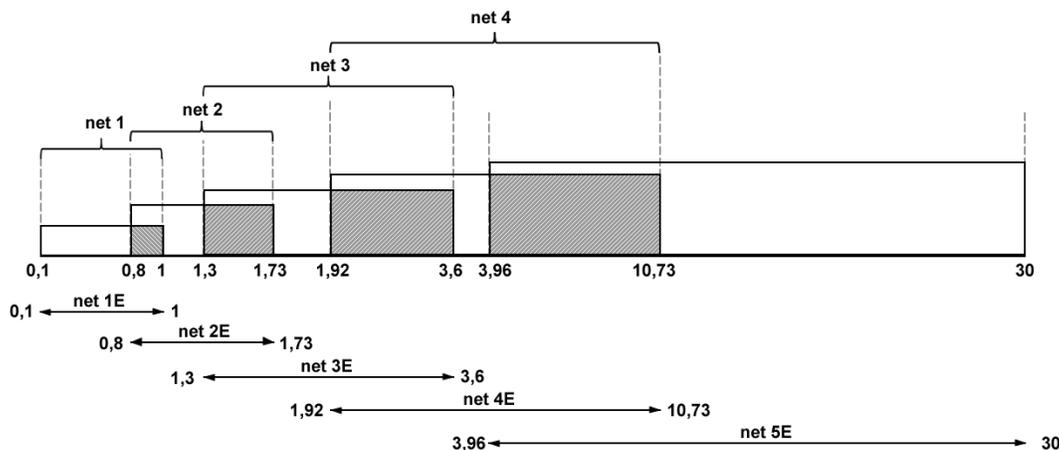


Fig. 5. Error Intervals for Stage One Networks and the Resulting Search Regions for Stage Two Networks.

and *net 4E* must be able to perform the estimation from $v = 1,92$.

The network *net 3* searches for groups of samples with $v < 2,7$ for sending them to *net 3E*. It has an error interval of $1,92 < v < 3,6$. Therefore, *net 3E* must be able to process samples up to $v = 3,6$ and *net 4E* must be able to perform the estimation from $v = 1,92$.

The network *net 4* searches for groups of samples with $v < 5,7$ for sending them to *net 4E*. It has an error interval of $3,96 < v < 10,73$. Therefore, *net 4E* must be able to process samples up to $v = 10,73$ and *net 5E* must be able to perform the estimation from $v = 3,96$. The network *net 5E* performs the estimation up to $v = 30$ which is the end of the search region.

D. B.3. Stage Two Networks

A double procedure was followed for training stage two networks. A subset was taken from Set C in the desired range, for example $0,1 < v < 1$ for *net 1E*, and multiple trainings were conducted to find the best network called *Best - C*. Afterwards, trainings were executed with a new network and a subset from Set A, calling *Best - A* to the best obtained network. Then the *Best - C* net was tested with Set A and the *Best - A* net was tested with Set C. The network with better performance facing the set for which it was not trained was chosen as the best.

The previously described process led to the selection of *Best - C* networks for *net 1E*, *net 2E* and *net 3E*. On the other hand, *Best - A* networks displayed the best outcomes for *net 4E* and *net 5E*.

E. Comparison of the Three Estimators

Once finished the design, the authors resolved to test the Two Stages Neuronal Solution with Set B, whose data were not used in the training of any of the nine networks included in the scheme. For comparing the results, they also tested the other two addressed estimators: the Method of Moments and the Single Neural Network.

As an intermediate test, the performance of the first stage was verified. For this purpose, the 20000 groups of samples from Set B were processed with the neuronal solution to check if the first stage was able to correctly choose the proper net from the second stage. The result was almost perfect: errors occurred only in four groups of samples corresponding to $v = 1.8310$, $v = 3.1599$, $v = 3.6774$ and $v = 3.8947$. This represents a success percentage of 99,98%. The same experiment was repeated with the MoM and it was found

that only in the 41% of the cases the estimator could choose the correct network from the second stage.

Regarding the performance test of the three estimators, Fig. 6 exhibits the mistakes made by each one in the search of the K shape parameter. After evaluating set B, it was found that the MoM has a mean error of 1,3249 and a maximum error equal to the maximum length of the search interval. Moreover, the Single Neural Network achieves an average error of 1,1399 and a maximum error of 19,9390. Finally, the Two Stages Neural Solution has an average error of 1,0716 and a maximum error of 19,77.

The above Fig.s reflect only a slight improvement in the accuracy achieved with the Two Stages Neural Solution. However, the true gain is visible in the graph for the $0,1 < v < 1$ region, also included in Fig. 6. Mistakes made in this region by the Two Stage Solution have very small magnitude. This is exactly the desired behavior, because the best estimate occurs in the region of greatest influence on the heavy property of the histogram tails and, therefore, in the region that requires a greater adjustment of the radar detector parameters.

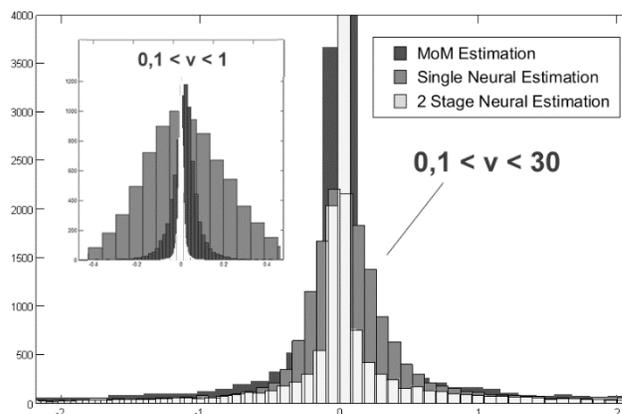


Fig. 6. Histogram of the Errors Made by the Three Estimators.

The exact Fig.s that characterize the estimators are given in Table III, where the results obtained by the Two Stages Solution are compared with the other two addressed estimators. For each region, the average error is offered first and the maximum error second. In addition, the last row contains the gain achieved by using the Two Stage Neural Solution in relation to the precision of the MoM estimates.

For example, the network *net 1E*, which operates in the $0,1 < v < 1$ region, sustains a mean error of 0,0095 and a maximum

TABLE III
MAGNITUDE OF MISTAKES MADE BY EACH ESTIMATOR IN FIVE DIFFERENT REGIONS.

Estimators/Estimation Region	0,1-1	0,8-1,73	1,3-3,6	1,92-10,73	3,96-30
MoM	0,0455-0,424	0,13-0,7	0,23-1,29	0,55-3,85	5,01-30
Single Neural Network	0,165-1,991	0,14-0,77	0,21-1,05	0,46-3,84	3,27-19,9
Two Stages Neural Net	0,0095-0,1076	0,04-0,23	0,11-0,7	0,4-3,03	3,33-19,7
Gain	4,79-3,94	3,25-3,04	2,1-1,84	1,37-1,27	1,5-1,52

error of 0,1076. That represents a 4,79 times better estimate than the one achieved by the MoM regarding the average, and a 3,94 times better estimate regarding the maximum error.

As it's shown in Table III, the gain of applying the Two Stage Solution is significant in the first three intervals; whereas the performance in the last two intervals is close to the one exhibited by the Single Neural Network. This means that the new solution is able to improve the performance in the region of interest, without affecting the quality of the overall estimate. Therefore, the superiority of the Two Stage Neuronal proposed Solution is demonstrated.

Moreover, Fig. 7 is presented as a complement of the information provided in Table III. The Fig. shows the error histograms for the last four regions.

F. About Speed of Execution

The execution speed of the algorithms is a critical issue in developments related to radar devices where a quick response is required. Indeed, the MoM using the second and fourth moments is very popular because it gets reduced to a closed mathematical expression that provides direct results, preventing the need for an iterative process. Then, the delay in the method's implementation lays only in the constant re-calculation of moments.

The Two Stages Neuronal Solution, despite containing nine neural networks, can be implemented without introducing significant delays. Although the response of each of the networks depends on the results of the others, all networks can evaluate information from the start. This can be done in parallel if implemented in FPGA (Field Programmable Gate Array) kits.

The evaluation will provide a result that can be manipulated using combinational logic, once the decision on which network must provide the estimation will become available. So, the neuronal solution does not require of any iterative process to produce and output. The only delay is introduced in the building of histograms, which will not take a longer time than the one required by the MoM for calculating the second and fourth order moments.

B. Significance of Results

Presented results support the neural networks capacity to solve statistical [39] and parameter estimation [40] problems. The current proposal applied a novel feature extraction method by using histograms of groups of K samples, which was combined with deep learning and classifier fusion approaches to produce an improved K parameter estimator. Recent investigations have also applied these principles to produce outstanding results in various fields [41-44]. So, the authors encourage new researchers to apply artificial intelligence methods for solving current problems.

IV. CONCLUSIONS AND FUTURE WORK

A new method for estimating the K shape parameter, based on a combination of nine independently trained neural networks, was created. The new method is as fast as the classic alternative based on the method of moments and achieves a much more accurate estimate in the region of small magnitudes of the K shape parameter. This is precisely the desired behavior for an estimator under K clutter, as the best performance is achieved in the region with the greater influence on the correction of the CFAR adjustment factor.

The new neural solution provides an important progress in in radar related solutions, as it allows the creation of a new improved CA-CFAR detector, able to adapt itself to changing sea conditions. The authors will focus next on the building of this new scheme which will improve the quality of detection by keeping the operational false alarm probability very close to the one conceived in the design. The authors will also focus on the development of a similar multi-stage neural solution for estimating the shape parameter under Log-Normal and Pareto clutter models.

To improve the presented scheme, the authors recommend adding an additional stage for further reducing the search regions before executing the final estimation. Additionally, the design could be improved by optimizing several simulation variables, such as the number of neurons in each hidden layer, the number of samples included in the histograms and the upper limit of the amplitude samples.

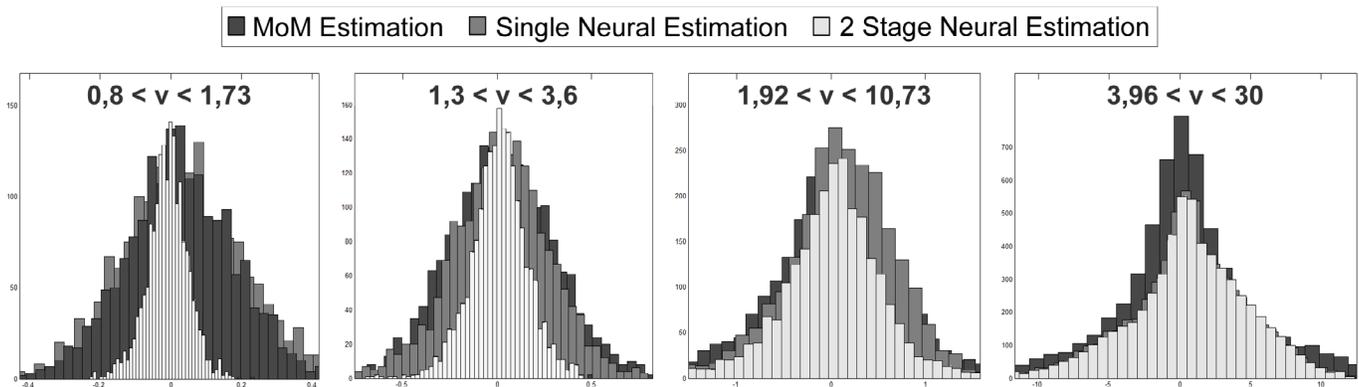


Fig. 7. Mistakes Made by each Estimator in Four Different Regions.

REFERENCES

- [1] D. K. Barton and S. A. Leonov, Radar Technology Encyclopedia (Electronic Edition) ed.: Artech House, 1998.
- [2] H. Meikle, Modern Radar Systems, 2nd Edition ed.: Artech House 2008.
- [3] M. I. Skolnik, Radar Handbook, 3er ed.: McGraw-Hill, 2008.
- [4] X. Shang and H. Song, "Radar Detection Based on Compound-Gaussian Model with Inverse Gamma Texture", IET Radar, Sonar and Navigation, vol. 5, pp. 315-321, 2010.
- [5] S. Ishii, S. Sayama, and K. Mizutani, "Effect of Changes in Sea-Surface State on Statistical Characteristics of Sea Clutter with X-band Radar", Wireless Engineering and Technology, vol. 2, pp. 175-183, 2011.
- [6] S. Sayama and S. Ishii, "Suppression of Log-Normal Distributed Weather Clutter Observed by an S-Band Radar", Wireless Engineering and Technology vol. 4 pp. 125-133, 2013.
- [7] K. Ward, R. Tough, and S. Watts, Sea Clutter Scattering, the K Distribution and Radar Performance, 2nd ed. London, United Kingdom: The Institution of Engineering and Technology, 2013.
- [8] Z. Zhijian, X. Ruilai, H. Yong, and G. Jian, "New Nonparametric Detectors under K-Distributed Sea Clutter in Radar Applications", in IEEE 2011 CIE International Conference on Radar, 2011, pp. 1752-1755.
- [9] X. Meng, G. Feng, H. Xue, and Z. He, "Wideband Radar Target Detection Theory in Coherent K Distributed Clutter", Research Journal of Applied Sciences, Engineering and Technology, vol. 5, pp. 1528-1532, 2013.
- [10] G. Tanriverdi, "Arma Model Based Clutter Estimation and its Effect on Clutter Supression Algorithms", Master of Science in Electrical and Electronics Engineering, The graduate School of Natural and Applied Sciences 2012.
- [11] R. C. Sánchez Rams, "Implementación del detector CA-CFAR en VHDL para crear un PC-Radar Cubano usando FPGA", Ing. en Telecomunicaciones y Electrónica, Departamento de Telecomunicaciones y Telemática, Facultad de Ingeniería Eléctrica, Instituto Superior Politécnico José Antonio Echeverría (ISPJAE), La Habana, Cuba, 2014.
- [12] J. R. Machado Fernández and J. C. Bacallao Vidal, "MATE-CFAR: Ambiente de Pruebas para Detectores CFAR en MATLAB", Telem@tica, vol. 13, pp. 86-98, 2014.
- [13] J. Z. Yim, C.-R. Chou, and W.-K. Wong, "A Study of the Statistics of Sea Clutter in the Northern Coast of Taiwan", in Proceedings of the Seventeenth International Offshore and Polar Engineering Conference, Lisbon, Portugal, 2007, pp. 1-6.
- [14] J. R. Machado Fernández, "Weibull Clutter Assumption Neural Adaptive Threshold Estimation Cell Averaging Constant False Alarm Rate (W-NATE-CA-CFAR) Detector (under revision)", Ingeniare, 2015.
- [15] J. R. Machado Fernández, J. C. Bacallao Vidal, and N. Chávez Ferry, "A Neural Network Approach to Weibull Distributed Sea Clutter Parameter's Estimation", Inteligencia Artificial vol. 18, pp. 3-13, 2015.
- [16] J. R. Machado Fernández and J. C. Bacallao Vidal, "Optimal Selection of the CA-CFAR Adjustment Factor for K Distributed Amplitude Samples with a Fluctuating Shape Parameter (Under Review)", Nova Scientia, 2016.
- [17] A. Machado Gil and B. García Delgado, "Reconocimiento de Parámetros Asociados a distribuciones del Clutter Marino con Redes Neuronales Artificiales", Ingeniero en Telecomunicaciones y Electrónica, Grupo de Radares, Departamento de Telecomunicaciones y Telemática, Instituto Superior Politécnico José Antonio Echeverría (ISPJAE), La Habana, Cuba, 2014.
- [18] A. Cetin, "CFAR Detection in K-Distributed Sea Clutter", Master of Science in Electrical and Electronics Engineering, The Graduate School of Natural and Applied Sciences of Middle East Technical University, 2008.
- [19] J. R. Machado Fernández and J. C. Bacallao Vidal, "Modelación de la Distribución K en MATLAB para Aplicaciones de Radar (bajo revisión)", Revista de Ingeniería Electrónica, Automática y Comunicaciones (RIELAC), 2016.
- [20] M. Greco, F. Bordonni, and F. Gini, "X-Band Sea-Clutter nonstationarity: Influence of Long Waves", IEEE Journal of Oceanic Engineering, vol. 29, 2004.
- [21] T. J. Nohara and S. Haykin, "Canadian East Coast Radar Trials and the K-Distribution", in IEE Proceedings on Radar and Signal Processing, 1991, pp. 80-88.
- [22] Y. Dong, "Distribution of X-Band High Resolution and High Grazing Angle Sea Clutter, Technical Report DSTO-RR-0316", Electronic Warfare and Radar Division, Defence Science and Technology Organization, Edinburgh, South Australia 2006.
- [23] R. S. Raghavan, "A Method for Estimating Parameters of K-Distributed Clutter", IEEE Transactions on Aerospace and Electronic Systems, vol. 27, pp. 238-246, 1991.
- [24] I. R. Joughin, D. B. Percival, and D. P. Winebrenner, "Maximum Likelihood Estimation of K distribution Parameters for SAR Data", IEEE Transactions on Geoscience and Remote Sensing, vol. 31, pp. 989-999, 1993.
- [25] D. Blacknell, "Comparison of Parameter Estimators for K-Distribution", IEE Proceedings on Radar, Sonar and Navigation, vol. 141, pp. 45-52, 1994.
- [26] P. Lombardo and C. J. Oliver, "Estimation of texture parameters in K-distributed clutter", IEE Proceedings on Radar, Sonar and Navigation, vol. 141, pp. 196-204, 1994.
- [27] D. R. Iskander and A. M. Zoubir, "Estimating the parameters of the K-distribution using the ML/MOM approach", Proceedings 1996 IEEE TENCON Digital Signal Processing Applications, TENCON'96, vol. 2, pp. 769-774, 1996.
- [28] D. R. Iskander and A. M. Zoubir, "Estimation of the parameters of the K-distribution using higher order and fractional moments [radar clutter]", IEEE Transactions on Aerospace and Electronic Systems, vol. 35 pp. 1453-1457, 1999.
- [29] W. J. J. Roberts and S. Furui, "Maximum likelihood estimation of K-distribution parameters via the expectation-maximization algorithm", IEEE Transactions on Signal Processing, vol. 48, pp. 3303-3306, 2000.
- [30] P. J. Chung, W. J. J. Roberts, and J. F. Bohme, "Recursive K distribution parameter estimation", IEEE Transactions on Signal Processing, vol. 53, pp. 397-402, 2005.
- [31] I. Antipov, "Analysis of sea clutter data, DSTO-TR-0647", Defence Science Technology Organisation, Australia 1998.
- [32] S. Watts, "Radar Detection Prediction in Sea Clutter using the Coumpound K-Distribution Model", in IEE Proceedings in Communications, Radar and Signal Processing, 1985, pp. 613-620.
- [33] D. Ling and W. Pingjun, "A Method for Determining Scale Factor of CFAR Detector Based on BP Neural Networks", presented at the 2nd International Conference on Computer Application and System Modeling, 2012.
- [34] J. A. Garzón Guerrero, "Clasificación de blancos de radar en ambientes de ruido arbitrario mediante resonancias naturales y técnicas de componentes principales", Doctor en Ciencias, Universidad de Granada, 2012.
- [35] N. B. Gálvez and J. E. Cousseau, "Improved Neural Network Based CFAR Detection for non Homogeneous Background and Multiple Target Situations", Latin American Applied Research, vol. 42, pp. 343-350, 2012.
- [36] K. Mezzoug and B. Djebbar, "Étude Comparative d'un Détecteur CFAR Neuronal de Plusieurs Cibles Radar dans un Fouillis de type K-Distribution", 2008.
- [37] A. Mezache and M. Sahed, "Parameter Estimation in K-Distributed Clutter with Noise using Nonlinear Networks", Université de Constantine, Faculté des Sciences de l'Ingénieur, 2010.
- [38] H. Demuth, M. Beale, and M. Hagan, Neural Network Toolbox 5: MathWorks, 2007.
- [39] N. Settouti, M. E. A. Bechar, and M. A. Chikh, "Statistical Comparisons of the Top 10 Algorithms in Data Mining for Classification Task", International Journal of Interactive Multimedia and Artificial Intelligence, Special Issue on Artificial Intelligence Underpinning, vol. 4, pp. 46-51, 2016.
- [40] C. K. Chong, M. S. Mohamad, S. Deris, M. S. Shamsir, Y. W. Choon, and L. En-Chai, "Improved Differential Evolution Algorithm for Parameter Estimation to Improve the Production of Biochemical Pathway", International Journal of Interactive Multimedia and Artificial Intelligence, Special Issue on Distributed Computing and Artificial Intelligence, vol. 1, pp. 22-29, 2012.
- [41] J. Singha and R. Hussain Laskar, "Recognition of global hand gestures using self co-articulation information and classifier fusion", Journal on Multimodal User Interfaces, vol. 10, pp. 77-93, 2016.
- [42] J. Singha and R. Hussain Laskar, "Self co-articulation detection and trajectory guided recognition for dynamic hand gestures", IET Computer Vision, 2015.
- [43] J. Singha and R. Hussain Laskar, "Hand gesture recognition using two-

level speed normalization, feature selection and classifier fusion”, *Multimedia Systems*, 2015.

- [44] V. B. Semwal, M. Kaushik, and G. C. Nandi, “Robust and accurate feature selection for humanoid push recovery and classification: deep learning approach”, *Neural Computing and Applications*, vol. 26, pp. 1-10, 2015.



José Raúl Machado Fernández received his Telecommunications and Electronics Engineering Degree from the Instituto Superior José Antonio Echeverría (ISPJAE-CUJAE) in 2012. He is currently a Ph.D. student at the same institution. His research topics include teledetection, digital signal processing, sea clutter modeling and the application of artificial intelligence for solving diverse engineering problems.



Jesús de la Concepción Bacallao Vidal received his engineering Telecommunications and Electronics Engineering Degree and Master Degrees from the Instituto Superior Politécnico José Antonio Echeverría (ISPJAE-CUJAE), and the Ph.D. Degree from the Instituto Técnico Militar José Martí in 2003. Since 2013, he has been co-directing the CUJAE radar research team. His research topics include teledetection, CFAR detectors performance evaluation and the creation of alternative detection schemes based on processing signal in the moment's domain (DRACEC).