ISSN 0027-1349, Moscow University Physics Bulletin, 2023, Vol. 78, Suppl. 1, pp. S188–S201. © Allerton Press, Inc., 2023.

MACHINE LEARNING IN NATURAL SCIENCES

Improvement of the AI-Based Estimation of Significant Wave Height Based on Preliminary Training on Synthetic X-Band Radar Sea Clutter Images

V. Yu. Rezvov[1,2*], M. A. Krinitskiy[1,2], V. A. Golikov[1,2], and N. D. Tilinina[1]

[1] *Shirshov Institute of Oceanology, Russian Academy of Sciences, Moscow, Russia*
[2] *Moscow Institute of Physics and Technology, Dolgoprudny, Moscow oblast, Russia*
Received August 21, 2023; revised October 5, 2023; accepted October 5, 2023

Abstract—Marine X-band radar is an important navigational tool that records signals reflected from the sea surface. Theoretical studies show that the initial unfiltered signal contains information about the sea surface state, including wind wave parameters. Physical laws describing the intensity of the signal reflected from the rough surface are the basis of the classical approaches for significant wave height (SWH) estimation. Nevertheless, the latest research claims the possibility of SWH approximation using machine learning models. Both classical and AI-based approaches require in situ data collected during expensive sea expeditions or with wave monitoring systems. An alternative to real data is generation of synthetic radar images with certain wind wave parameters. This Fourier-based approach is capable of modelling the sea clutter images for wind waves of any given height. Assuming a fully-developed sea, we generate synthetic images from the Pierson–Moskowitz wave spectrum. After that, we apply an unsupervised learning using synthetic radar images to train the convolutional part of the neural network as the encoding part of the autoencoder. In this study, we demonstrate how the accuracy of SWH estimation based on radar images changes when the neural network is pretrained on synthetic data.

Keywords: wind waves, marine radar, sea clutter, significant wave height, Pierson–Moskowitz spectrum, synthetic radar images, preliminary training

DOI: 10.3103/S0027134923070275

1. INTRODUCTION

Climate change is one of the most serious problems of the modern world. In particular, it causes local precipitation patterns and local changes in wind speed [1]. Climate change studies require description of physical processes and interactions in the atmosphere-ocean system. These sea-wave interactions strongly depend on surface wind speed and wind waves. For example, long-term climate reconstructions, such as Global Atlas of the Ocean Waves [2], consume different ocean wave parameters.

Marine X-band radars are important for ship navigation and safety because they detect obstacles. However, raw radar images of the sea clutter also contain meaningful information. Spatial distribution of the reflected signal allows deriving characteristics of wind waves and swell. Furthermore, with radar images, we can estimate significant wave height (SWH), which is one of the most important sea surface values.

The classical method uses Fourier analysis and linear dispersion relationship to detect the wave signal on the time series of radar data. The necessity of modulation transfer function and calibration coefficients specific for each radar antenna [3] limits the generalizing ability of this methodology. Despite this fact, the classical method is widely used for real-time approximation of ocean wave parameters from the back-scatter spectrum of radar images [4, 5].

Besides classical approach, radar data processing also includes methods that have a potential to be faster and more independent of radar antenna. Some publications also demonstrate higher quality of SWH estimation in contemporary AI-based techniques [6] compared to classical approaches. In the machine-learning group of methods, the functional relationship between a radar image and the appropriate SWH is approximated by training a model on a massive

*E-mail: rezvov.vyu@phystech.edu

Table 1. Research expeditions

Expedition	Departure	Arrival	No. of Spotter buoy locations	No. of SeaVision locations
ASV50	Kaliningrad, Russia, 7 Aug 2020	Arkhangelsk, Russia, 13 Sep 2020	24	157
AI57	Kaliningrad, Russia, 25 Jun 2021	Arkhangelsk, Russia, 21 Jul 2021	12	76
AI58	Kaliningrad, Russia, 10 Aug 2021	Kaliningrad, Russia, 9 Sep 2021	16	55
AI63	Arkhangelsk, Russia, 29 Sep 2022	Arkhangelsk, Russia, 7 Dec 2022	30	209
Total			**82**	**497**

dataset. Consequently, approximation quality and generalization ability heavily rely on dataset size and distribution.

The authors of [7] solve the problem of insufficient data with the help of an algorithm of generating synthetic radar images with any given SWH. A similar approach is based on the Fourier synthesis of a realistic ocean surface [8]. The artificial sea surface is processed to obtain a realistic X-band radar image. Although [7] does not implement image synthesis as part of a machine-learning approach, we suppose that a combination of an artificial neural network and a synthetic dataset is able to increase the model's quality.

Convolutional neural networks (CNNs) demonstrate their efficiency in image recognition and feature extraction across different branches of science, including geosciences. For example, [9] shows a CNN application for real-time SWH estimation from a series of real ocean photo images.

Summing up, recent advances in artificial neural networks and the emergence of contemporary machine-learning techniques have led to a significant increase in the variety of radar image processing methods. However, the results of research on this topic are contradictory, and several questions remain relevant. Therefore, the assessment of contemporary methods for radar image generation and SWH estimation is still a task of current scientific interest.

In this paper, we present an application of the convolutional neural network to the dataset of real marine radar images of sea clutter to estimate significant wave height. We preliminarily train an AI-based autoencoder on synthetic radar images. After that, we use the pre-trained encoder as a part of the proposed CNN.

The paper is organized as follows. In Section 2 we provide the details of the real radar image dataset and the methodology of the radar image synthesis. Section 3 describes the architecture of the applied neural network, quality metrics and training and evaluation procedures. In Section 4, we provide the results and their analysis. Concluding remarks are made in Section 5.

2. MATERIALS AND METHODS

2.1. Initial Data

In this study, we adhere to the methodology of data collection from [5]. We use data samples from four research expeditions in the Atlantic and Arctic oceans, carried out by the Shirshov Institute of Oceanology of the Russian Academy of Sciences within the governmental program of regular ocean observations.

The expeditional pathways contain points of SeaVision and Spotter buoy measurements. We provide general information about research cruises in Table 1. The locations of sea wave observations were determined by local weather conditions and time availability. Hereafter, we define "stations" as the locations of wind wave observations.

For each station we collect simultaneous in situ observations, X-band radar signal as sea clutter images, and/or Spotter wave buoy (`https://www.sofarocean.com/products/spotter`) measurements. In this study, we only use stations that provide data from both the Spotter buoy and SeaVision at the same time despite the fact that the whole dataset contains locations with only SeaVision measurements (Table 1). Availability of Spotter-SeaVision data pairs is necessary for proper learning of artificial neural networks.

The Spotter wave buoy provides highly accurate estimations of wind wave characteristics and collects the training dataset of SWH. SeaVision provides one sea clutter image per two seconds.

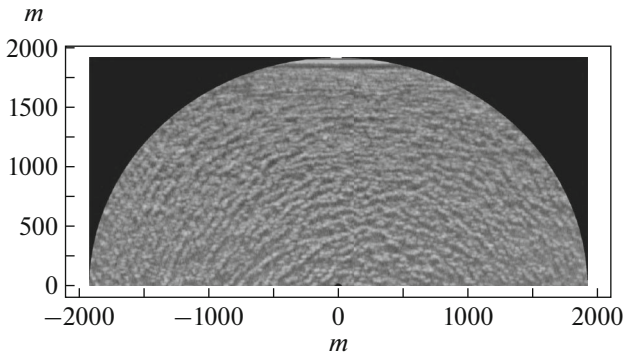

Fig. 1. An example of a preprocessed radar image from the AI57 marine research mission of the Institute of Oceanology of the Russian Academy of Sciences. Here, the magnitude of the back-scattered radar signal is presented in grayscale. The position of the shipborne navigation X-band radar is at (0, 0) in these coordinates.

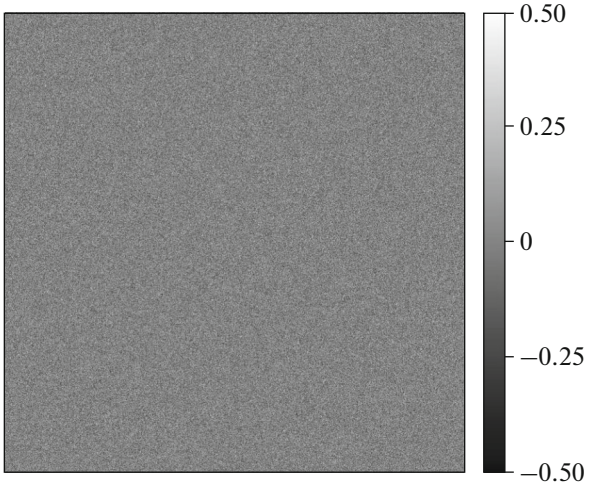

Fig. 2. White-noise-generated image as the first step of the algorithm for simulating a radar image of sea clutter.

The center of the digitized sea clutter images coincides with the location of the ship. Images cover >7 km radius around the ship with the spatial resolution of 1.875 m. The image contains a signal of variegated structure and intensity caused by local wind direction, ship rotation and the electromagnetic signal reflection from the rough ocean surface (e.g., [10]).

Each image has an area where the wave field is expected to provide the most distinct signal. Images also have a "blind" zone near the center due to signal reflection from the ship. We exclude the part of the image within a 30-m radius around the ship to avoid this effect. Then, we suppose that a 3500 m radius ensures the highest significant variability observed in radar data. As a result, we choose the 30–3500 m range for further processing. As the distance from the ship increases, the reflected signal becomes less distinct; thus, we select only the first 1024 pixels for every direction, equal to a 1920 m radius from the radar antenna.

One of the important steps in data preprocessing is the choice of the optimal 180° sector containing pure wave signal. The algorithm used in this paper extracts the most contrasting area from the entire radar image to provide the most clearly identified wind waves. For every station, the sector with the largest standard deviation over time is the optimal sector. Hence, in this study, considering the spatial resolution of 1.875 m, we work with 2048 × 2048 pixels images, in contradistinction to 384 × 384 in [4]. An example of the preprocessed radar image is shown in Fig. 1.

2.2. Synthesis of Sea Surface Images

We elaborate on the methodology of generating realistic synthetic radar images based on [7] and realistic ocean scenes from [8]. The authors of [8] developed a technique to present a fully developed sea using an empirical modified Pierson–Moskowitz sea spectrum [9].

According to [8], we first generate white-noise images with uniformly distributed noise having intensities from −0.5 to 0.5. We assume that the spatial resolution of the gray image is 1.875 m, equal to that of the real radar images. The result of this step is a 2088 × 2088 pixels image of random gray shades (see Fig. 2).

After that, we perform the two-dimensional forward Fourier transform of the image to generate an array of complex numbers. The magnitude of these complex Fourier components is shown in Fig. 3.

In this research, we assume fully developed wind waves. Thus, the wave spectrum remains con-

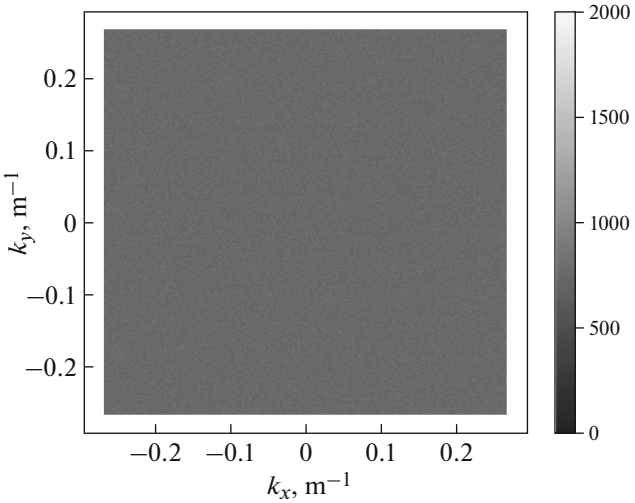

Fig. 3. Fourier transform magnitude of the white-noise image from Fig. 2.

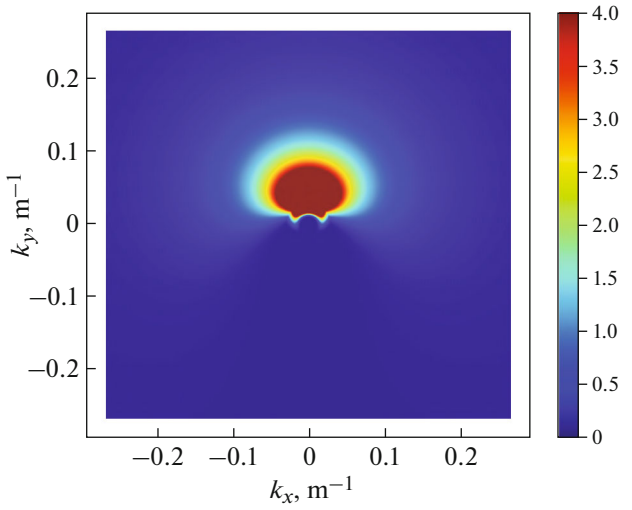

Fig. 4. Pierson–Moskowitz spectral filter for white-noise Fourier components.

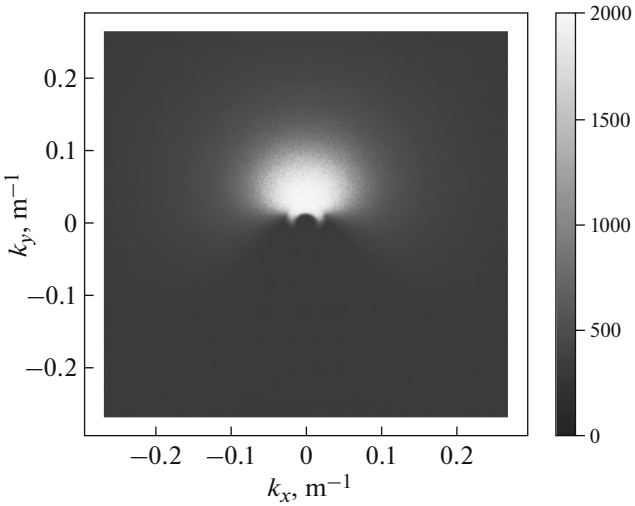

Fig. 5. Filtered magnitudes of white-noise Fourier components from Fig. 3.

stant. Under these assumptions, W. Pierson and L. Moskowitz mathematically modeled the downwind power spectrum $F_{\mathrm{PM}}(f)$ [9]:

$$F_{\mathrm{PM}}(f) = \frac{\alpha g^2}{(2\pi)^4 f^5} \exp\left[-\frac{5}{4}\left(\frac{f_m}{f}\right)^4\right], \quad (1)$$

where f is the frequency [Hz], f_m is the peak frequency [Hz], $\alpha = 0.0081$ is the Phillips constant, and g is the gravitational constant.

We can immediately obtain f_m from 10 m surface wind speed U_{10} [9]:

$$f_m \approx 0.13\frac{g}{U_{10}}. \quad (2)$$

By definition, the significant wave height is the mean wave height of the highest third of the waves in the wave record: $SWH = \frac{1}{N/3}\sum_{j=1}^{N/3} H_j$. For a given power spectrum $S(\omega)$, the significant wave height: $SWH = 4\sqrt{\int_0^{+\infty} S(\omega)\, d\omega}$. Thus, from (2), for the Pierson–Moskowitz spectrum (1), we obtain:

$$SWH \approx 0.22\frac{{U_{10}}^2}{g}. \quad (3)$$

The importance of (1) is that it depends on only one parameter—the peak frequency f_m. Moreover, the surface wind U_{10}, the significant wave height SWH and the peak frequency f_m strictly depend on each other. Thereby, we can calculate the one-dimensional, in the direction of the wind, fully developed spectrum from (1) using either U_{10} or SWH (3). To take into account the wind direction, the authors of [11] suggested a two-dimensional spectrum $F(f, \varphi)$:

$$F(f, \varphi) = F_{\mathrm{PM}}(f) D(f, \varphi), \quad (4)$$

where $F_{\mathrm{PM}}(f)$ is the one-dimensional Pierson–Moskowitz spectrum and $D(f, \varphi)$ is a normalized directional multiplier weighting the spectrum at angle φ from the downwind direction. According to [11]:

$$D(f, \varphi) = \frac{\Gamma^2(p+1)}{2^{1-2p}\pi\Gamma(2p+1)}\left[\cos\left(\frac{\varphi}{2}\right)\right]^{2p}, \quad (5)$$

where

$$p = 9.77\left(\frac{f}{f_m}\right)^{\mu}$$

and

$$\mu = \begin{cases} 4.06, & f \le f_m \\ -2.34, & f > f_m. \end{cases}$$

The example of the normalized filter $F_{\mathrm{PM}}(f)D(f, \varphi)$ for a wind speed of 15 m/s is shown in Fig. 4.

We multiply two-dimensional spectrum of fully developed wind waves (4) by the magnitudes of the Fourier components of the initial white-noise image.

Our two-dimensional white-noise spectrum creates a narrower profile near f_m (2) in the downwind direction of the spectrum and forms a bimodal spectrum shape for φ near 90° from downwind. The resulting spectrum suppresses the long-crested peak frequency components, while retaining the nonpeak frequency [8]. The filtered magnitudes of the white-noise Fourier components from Fig. 3 are shown in Fig. 5.

Next, we combine the filtered magnitudes and the original phase of the white-noise Fourier components. After that, we use the inverse Fourier transform of the filtered components to generate a 2088 ×

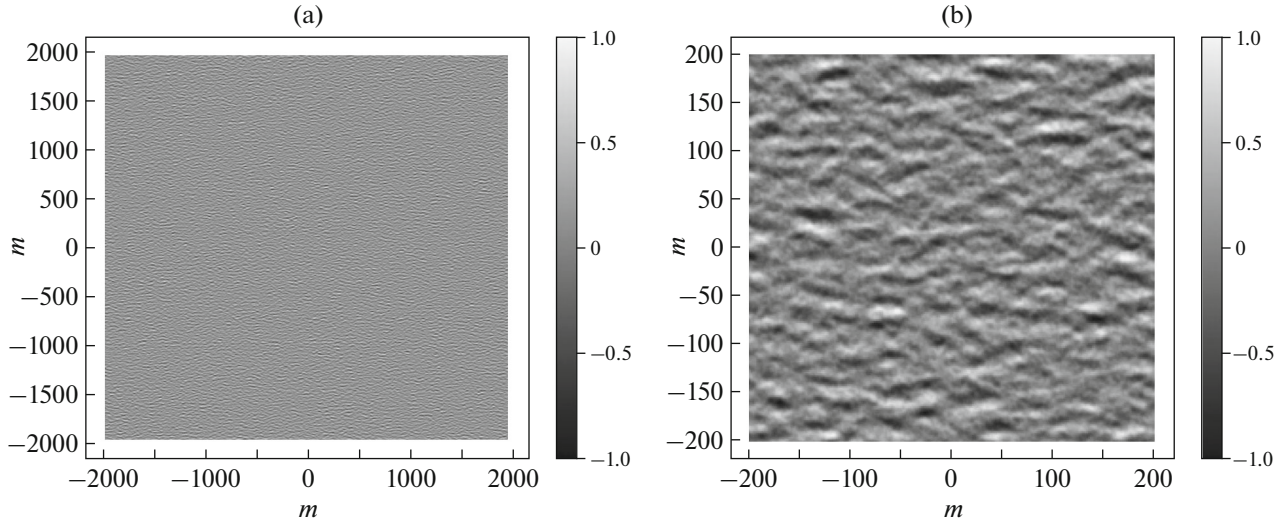

Fig. 6. The synthetic ocean image created by processing the white-noise image from Fig. 2: (a) full image; (b) 200 m × 200 m area.

2088 array of complex numbers [8]. The real part of this array is a synthetic sea surface image. The realistic result on different scales is shown in Fig. 6.

2.3. Synthesis of Radar Images

We convert sea surface images obtained in Subsection 2.2 to radar images.

First, from the 2088 × 2088 array of the sea surface, we select points that correspond to distances on the real radar images described in Subsection 2.1. The resulting "radar area" is shown in Fig. 7.

After that, we include two main simple geometrical effects that influence the X-band signal reflected from the sea surface: shadowing and tilt modulation [7]. Then, we shortly summarize the model proposed by [12] modified by [7].

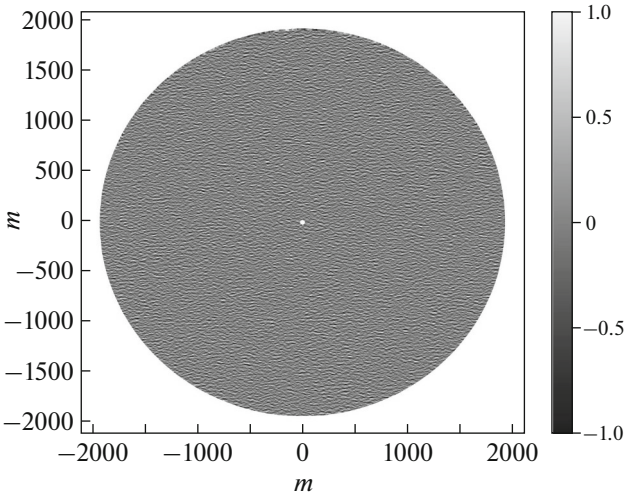

Fig. 7. Synthetic sea surface in the radar image area.

We describe the shadowing effect using the geometrical optics approximation. In some areas, the sea surface prevents the reflection of radar rays from other parts of the surface and causes shadowing of the nearby waves. Consequently, the radar antenna does not receive any signal from the shadowed parts of the sea surface [7]. Obviously, this phenomenon depends on the grazing angle and the antenna height Z_a, so it becomes more severe with increasing distance from the radar (Fig. 8).

For simplification, we assert that the shadowed area signal is equal to zero. Hence, the coefficient of the synthetic image shadowing is given by the equation below:

$$M_{\text{sh}}(\vec{r}) = \begin{cases} 0, & \text{if} \quad \vec{r} \in \text{shadowed area} \\ 1, & \text{otherwise.} \end{cases}$$

The result of the shadowing effect is $H(\vec{r})M_{\text{sh}}(\vec{r})$, where $H(\vec{r})$ is a sea surface field.

In the case of tilt modulation, the steepness of the observed surface slope affects the power amplitude received by the radar antenna [7]. Thus, the received back-scattered signal is modulated by the angle θ between the radar illumination ray $\vec{u}(\vec{r})$ and the normal vector $\vec{n}(\vec{r})$ to the wave surface (Fig. 8). The antenna receives the signal modulated by the coefficient $M_{\text{tilt}}(\vec{r}) = \vec{u}(\vec{r}) \cdot \vec{n}(\vec{r})$. Since $\vec{u}(\vec{r})$ and $\vec{n}(\vec{r})$ are vectors of length 1, the tilt modulation coefficient for every point $\vec{r}$ on the sea surface is:

$$M_{\text{tilt}}(\vec{r}) = \begin{cases} 0, & \text{if} \quad \vec{r} \in \text{shadowed area} \\ \cos\theta, & \text{otherwise.} \end{cases} \qquad (6)$$

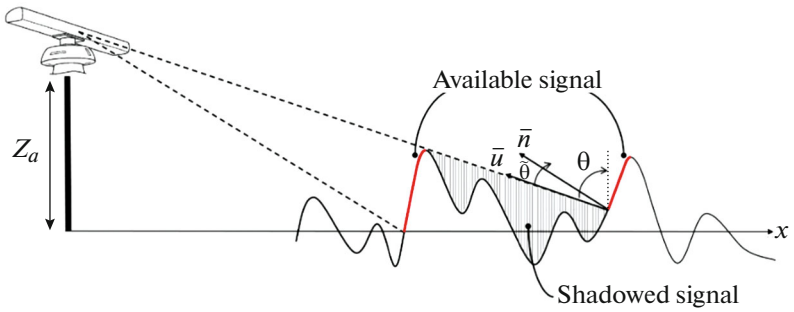

Fig. 8. Geometrical scheme of shadowing and tilt modulation effects [7].

Summing up, the resulting amplitude of the back-scattered signal on a synthetic radar image is $M_{\text{sh}}(\vec{r})M_{\text{tilt}}(\vec{r})$.

2.4. Dataset of Synthetic Images

In our study, we generate a set of 10 000 radar images.

First, we generate 10 000 values of the U_{10} speed, which is directly connected with the SWH value (3) and the power spectrum of wind waves (2), (5). The wind speed is distributed uniformly from 3 to 20 m/s to ensure proper learning quality.

After that, for every wind speed value, we generate a corresponding 2088×2088 image, as described in Subsection 2.2. Then, for every point in the range from 30 m to 1920 m from the center of the sea surface field, we compute the shadowing effect coefficient. This 30–1920 m range coincides with that of real radar images. The example of the shadowing effect for the sea surface from Fig. 7 is shown in Fig. 9.

For the points outside the shadowed areas, we compute the grazing angles assuming that the radar antenna height $Z_a = 20$ m. As described above, in Subsection 2.3, we evaluate angle θ and, as a result, tilt modulation effect $M_{\text{tilt}}(\vec{r})$. The example of this effect for the sea surface from Fig. 7 is shown in Fig. 10.

To exclude edge effects, we remove points from every side of the synthesized radar image resulting in a 2048×2048 array of the back-scattered signal amplitude. We divide this square image into two halves with a 1024×2048 size. We choose the direction of the image incision so that the first half corresponds to the downwind direction, and the second half corresponds to the upwind direction. The resulting shape is equal to that of real data.

This procedure leads to 20 000 samples of synthetic radar images. The examples for different values of U_{10} wind speed are in Fig. 11.

3. AI-BASED MODELS

3.1. Baseline Model

As an example of a deep-learning model, we use a convolutional neural network (CNN). CNNs are a

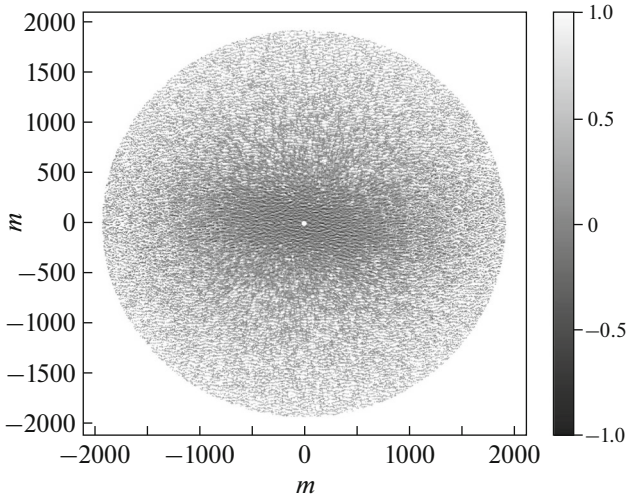

Fig. 9. Shadowing effect for wave height field from Fig. 7.

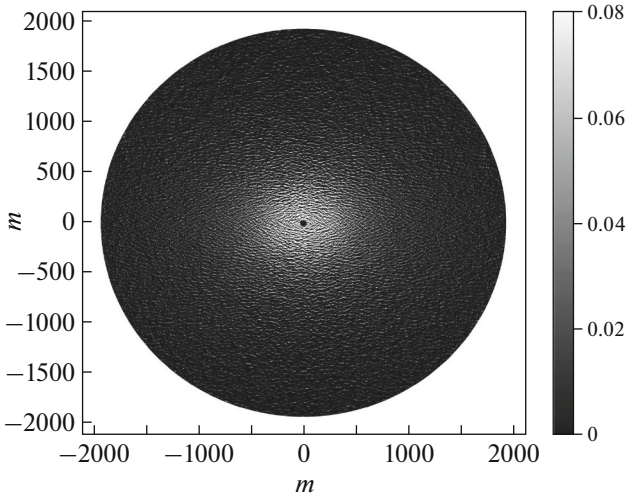

Fig. 10. Tilt modulation $M_{\text{tilt}}(\vec{r})$ for wave height field from Fig. 7.

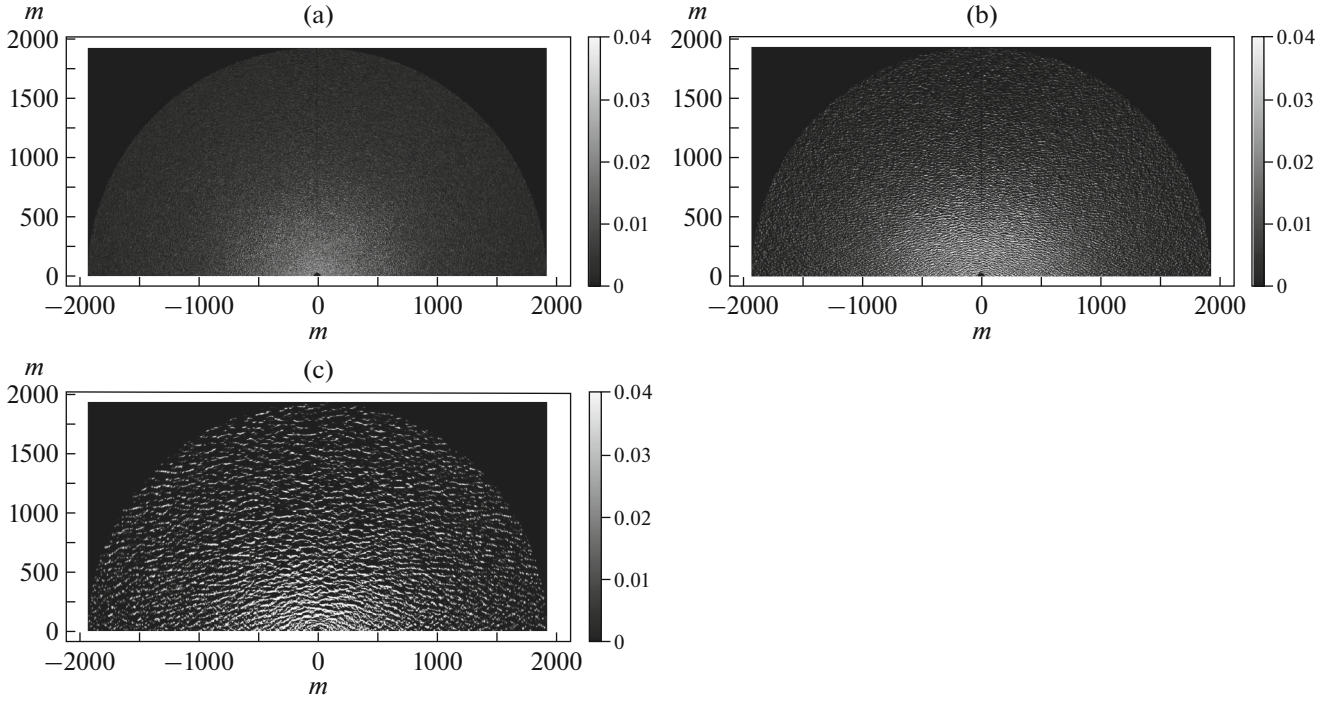

Fig. 11. Examples of synthetic radar images with different U_{10} wind speeds: (a) 4; (b) 12; (c) 20 m/s.

kind of artificial neural networks (ANNs). ANNs, in short, are parametric mappings that optimize model parameters during the model training. This approach identifies various abstract features of the input data. Particularly, in CNNs, a fixed-size convolution kernel is sequentially applied to regular input data. For each convolutional layer, the convolution kernel is a set of trained parameters in the form of an array [13].

Increasing CNN depth is expected to improve the quality of output prediction. However, such an increase leads to training instability of the back-propagation algorithm. Learning becomes inefficient due to "vanishing gradients." This negative effect accumulates extremely small gradients of model parameters. As a result, the product of the gradient vector and the learning rate coefficient tends to zero, and the parameters updated at each optimization step remain constant. The effect of "vanishing gradient" remains a significant problem for CNNs [13].

One of the effective ways to solve the problem of learning instability is to add connections that skip the intermediate layers of the model. These connections reduce the risk of accumulating small gradients. An example of this is residual connections, which add the output from one intermediate layer to the output of a subsequent layer. Residual connections allow the model to use outputs from the initial layers at the beginning of the network [13].

We combine the advantages of deep CNNs and residual connections in the modified ResNet-50, based on [14]. In our study, CNN obtains SWH value from a real 1024×2048 radar image. The proposed network includes an input layer for extracting raw features, followed by a sequence of convolutional layers and residual blocks for processing the extracted features. Unlike the original ResNet-50, we replace ReLU activation with Mish activation [15]:

$$\mathrm{Mish}(x) = x \tanh\left[\mathrm{Softplus}(x)\right].$$

The input is not a 3-channel RGB image, so we change the number of input and output channels of the first convolutional layer from 3 to 1 and from 64 to 128, respectively. We also change the fully-connected (FC) layer between the last convolution and the output. The input of this layer is now a vector with a length of 2048. The output of the modified ResNet-50 is one number which is learnt to be equal to SWH for this radar image. The outline of the modified ResNet-50, along with its residual blocks, is shown in Fig. 12.

3.2. Autoencoder

Autoencoders are an example of unsupervised machine learning. They are ANN models consisting of

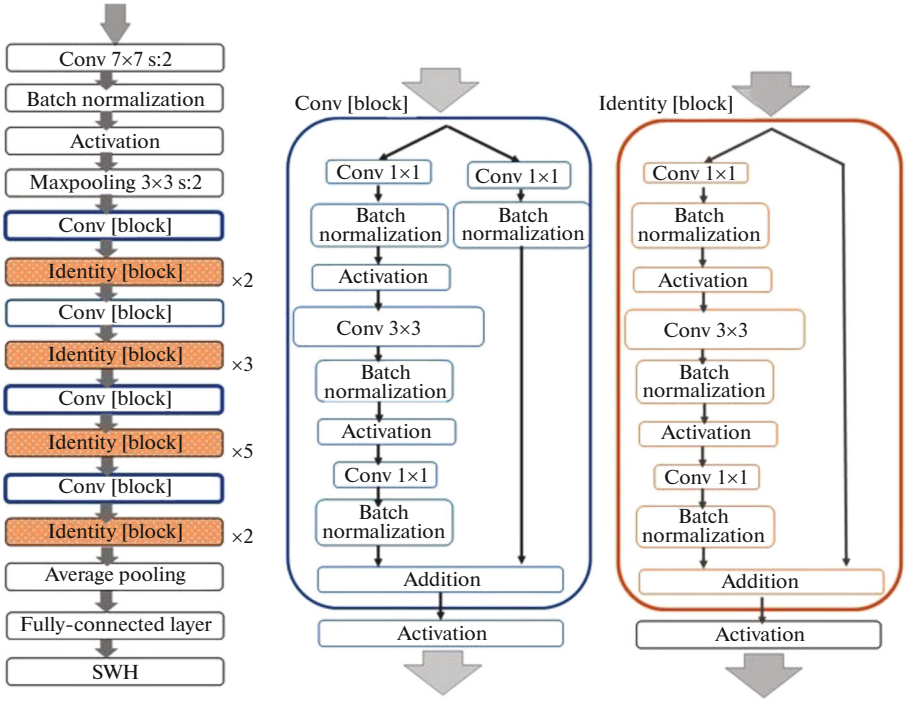

Fig. 12. Outline of the modified ResNet-50 architecture [16].

an encoder and a decoder. The encoder converts complex input features into a lower number of dimensions from the unlabeled data [17]. The result of the decoder is a latent vector that is a vector in a low-dimensional latent space. Thus, the last layer of the encoder is also referred to as a "bottleneck layer". Then, the decoder tries to reconstruct the original complex input from the simpler encoded features in the latent space [18].

In this work, the autoencoder is used as a model for pretraining. The weights of the trained encoder are then applied to the training of the modified ResNet-50. The layer-by-layer description of our autoencoder is shown in Table 2.

Our convolutional encoder starts with an input synthetic image of a 1024×2048 size. The synthetic and the real radar images are shape-like. Consequently, for the encoder, we use the modified ResNet-50 without its last FC layer—"convolutional part of the modified ResNet-50." The output of the encoder is a latent vector with a length of 2048. The bottleneck layer is a FC layer increasing the number of channels from 2048 to $32 \times 32 \times 32 = 32\,768$ channels. This is required for a proper further increase in spatial dimensions.

After the bottleneck layer, we build a sequence of five convolutional blocks. Every convolutional block consists of upsampling operation, two convolutional layers, batch normalization for learning stability, and the Mish activation. We employ a special upsampling operation, namely, PixelShuffle, presented in [19] as an efficient and flexible, yet trainable, alternative for the upsampling procedure instead of the interpolation-based upsampling and transposed convolution. In general, *PixelShuffle* rearranges elements in a tensor of shape $(C \times r^2, H, W)$ to a tensor of shape $(C, H \times r, W \times r)$, hence, this function requires a sufficient number of input channels. Nevertheless, we consider $PixelShuffle$ to be more effective and appropriate than $Upsample$ function, because $Upsample$ function, especially in the beginning of the decoder, causes "checkerboard" artifacts [20].

The output of five convolutional blocks is $4 \times 512 \times 512$. By definition, $PixelShuffle$ preserves the ratio of the spatial dimensions, therefore, we add $Upsample$ in convolutional blocks nos. 6 and 7 to convert a square image to a rectangular one with minimal loss of quality. Negative effects of $Upsample$ are also minimized by a bigger kernel size 7×7 (convolutional block no. 6) and 9×9 (convolutional block no. 7). The output of the decoder is an image, with the same 1024×2048 size as it is for the input.

3.3. Quality Metrics

We consider various quality metrics to compare models and track learning properly. For the autoencoder, we use the simplest quality metrics—the root mean-squared error. We refer to it as $RMSE_{AE}$.

Table 2. Autoencoder outline

Autoencoder component	Layers	Output size
Encoder	Convolutional part of modified ResNet-50	[2048, 1, 1]
Bottleneck layer	Fully-connected layer (2048 → 32 × 32 × 32 channels)	[32 768, 1, 1]
Decoder	• PixelShuffle (upscale factor = 32) • 3 × 3 convolutions, 32 → 128 → 64 channels • Batch normalization + Mish activation	[64, 32, 32]
	• PixelShuffle (upscale factor = 2) • 3 × 3 convolutions, 16 → 64 → 32 channels • Batch normalization + Mish activation	[32, 64, 64]
	• PixelShuffle (upscale factor = 2) • 3 × 3 convolutions, 8 → 32 → 16 channels • Batch normalization + Mish activation	[16, 128, 128]
	• PixelShuffle (upscale factor = 2) • 3 × 3 convolutions, 4 → 16 → 8 channels • Batch normalization + Mish activation	[8, 256, 256]
	• PixelShuffle (upscale factor = 2) • 3 × 3 convolutions, 2 → 8 → 4 channels • Batch normalization + Mish activation	[4, 512, 512]
	• Upsample (512, 512) → (768, 1024) • 7 × 7 convolution, 4 → 2 channels • Batch normalization + Mish activation	[2, 768, 1024]
	• Upsample (768, 1024) → (1024, 2048) • 9 × 9 convolution, 2 → 1 channels	[1, 1024, 2048]

This type of the reconstruction error is the difference between the input data and the data after the autoencoder compresses and decompresses the input. $RMSE_{AE}$ evaluates the noise added to the input by running through the autoencoder.

If we denote our autoencoder model as AE and a batch of input images as y, then the output batch is $y^* = AE(y)$. We compute $RMSE_{AE}$ summarizing only by points that are within the radius range from 30 to 1920 m to exclude nonmeaningful areas of the radar image. The number of points in the input batch is W_{batch}, and the result is:

$$RMSE_{AE} = \sqrt{\frac{1}{W_{\text{batch}}} \sum_{i,j,k} (y^*_{i,j,k} - y_{i,j,k})^2}. \quad (7)$$

In (7), $y_{i,j,k}$ and $y^*_{i,j,k}$ are the identical elements of the input and the output tensors. We summarize the difference between the input and the output point-by-point. The indices i, j, and k are in ranges of respective spatial dimensions and batch size, taking into account a 30–1920 m mask.

It is shown that for images, RMSE fails to reveal defective regions that have been visually altered when intensity values stay roughly consistent [21]. In this study, we use a perceptual quality metric based on structural similarity that checks interdependences between local image regions. In contrast to comparing pixel values in RMSE, structural similarity also considers contrast and structural information. The structural similarity index measure (SSIM) is computed as follows:

$$SSIM(y, y^*) = \frac{(2\mu\mu^* + C_1)(2\sigma_{yy^*} + C_2)}{(\mu^2 + \mu^{*2} + C_1)(\sigma^2 + \sigma^{*2} + C_2)},$$

where μ is the pixel sample mean of y, μ^* is the pixel

sample mean of y^*, σ^2 is the variance of y, σ^{*2} is the variance of y^*, σ_{yy^*} is the cross-correlation of y and y^*. C_1 and C_2 are constant values equal to 0.01^2 and 0.03^2, respectively.

In this paper, SSIM is measured between two windows of the size 128×128 pixels, applied to the input and the output images. SSIM is between -1 and 1. The higher SSIM indicates higher similarity.

For the modified ResNet-50, we use only RMSE because the output of this CNN is a vector of significant wave height values with the length of batch size BS:

$$RMSE = \sqrt{\frac{1}{BS}\sum_{k=1}^{BS}(SWH_k^* - SWH_k)^2},$$

where SWH is a true value measured by the Spotter buoy (Subsection 2.1), and SWH^* is the output value of the corresponding real radar image.

3.4. Model Training and Evaluation

First, we train the autoencoder. We randomly split the dataset of 20 000 synthetic radar images into three parts: training dataset (14 000 images), validation dataset (3000 images), and testing dataset (3000 images). During training and evaluation, we load images into our model in the batches of 2 on account of memory restriction. The random split provides the uniform distribution of modeled wind wave conditions in all three datasets. As described above, the input is acquired as a batch of single-channel gray-scale images.

We train the autoencoder with point-by-point mean-squared-error (MSE) loss, excluding points out of the radius range from 30 to 1920 m. The masked MSE allows training only areas with meaningful radar information. All the parameters both in the encoder, and in the decoder are trained. The proposed autoencoder network is trained for 100 epochs using the ADAM [22] optimizer with an initial learning rate of 0.05. We set the learning rate of each parameter group using a cosine annealing schedule, where the maximum learning rate is equal to the initial learning rate and the minimum learning rate before the restart equal to 1×10^{-6}. The restarts are on epochs 25 and 75. In every restart, the learning rate decay is 0.75.

After every training epoch, we evaluate the reconstruction quality with $RMSE_{AE}$ and SSIM on the validation dataset (Subsection 3.3). We stop the autoencoder training after epoch 100 before the restart on epoch 150. We suppose that training is saturated because $RMSE_{AE}$ and SSIM remain almost constant. After training, we provide final evaluation on the testing dataset with the same quality metrics.

For the modified ResNet-50, we also split the dataset of the real images into three parts: the training dataset (70%), the validation dataset (15%) and the testing dataset (15%). The batch size is 20 for both training and evaluation procedures. The training loss for experiments with the modified ResNet-50 is simple MSE loss.

The first experiment on this architecture is a baseline model—we train the whole modified ResNet-50 without any pretrained weights. In this case, the CNN is trained for 64 epochs using the ADAM [22] optimizer with an initial learning rate of 1×10^{-4}. We set the cosine annealing schedule with the maximum learning rate equal to the initial learning rate and the minimum learning rate equal to 5×10^{-7}. The restarts should have been on epoch 128 but the training is saturated after epoch 64. After every training epoch, we evaluate SWH regression quality with $RMSE$ on the validation dataset (Subsection 3.3). After training, we provide final evaluation on the testing dataset with the same quality metric.

Before the second experiment, we save the parameters of the encoding part of the autoencoder (see Table 2). After that, we load the saved weights into the convolutional part of the modified ResNet-50. We train ResNet-50 for 40 epochs with the "frozen" convolutional pretrained weights. Thus, the first 40 epochs are aimed at fully-convolutional layer learning. The optimizer is ADAM [22] with an initial learning rate of 1. We set the cosine annealing schedule with the maximum learning rate equal to the initial learning rate and the minimum learning rate equal to 1×10^{-4}. The learning rate restarts at epoch 20 with 0.75 decay.

After learning with the pretrained convolutional part, we "unfreeze" convolutional weights, restart the learning rate and train the "unfreezed" model for 140 epochs. After every training epoch, we evaluate SWH regression quality with RMSE on the validation dataset and after epoch 180 we finally evaluate the model on the testing dataset.

4. RESULTS AND DISCUSSION

4.1. Autoencoder Reconstruction

Table 3 shows the results of the autoencoder training.

For 100 epochs, the autoencoder is trained from 0.824 to 0.767 in $RMSE_{AE}$ and from -0.025 to 0.090 in SSIM. The validation dataset shows a similar trend—$RMSE_{AE}$ decreases from 0.782 to 0.768 and SSIM decreases from -0.107 to 0.116. Noticeably, SSIM and $RMSE_{AE}$ are weakly correlated

Table 3. Autoencoder training and evaluation

Epoch	Training dataset		Validation dataset	
	$RMSE_{AE}$	$SSIM$	$RMSE_{AE}$	$SSIM$
1	0.824	−0.025	0.782	−0.107
25	0.772	0.071	0.763	0.070
50	0.776	0.039	0.770	0.050
75	0.768	0.049	0.770	0.027
100	0.767	0.090	0.768	0.116
			Testing dataset	
			0.775	**0.151**

and decrease nonsimultaneously. We explain it by different characteristic scales of these two quality metrics. $RMSE_{AE}$ is very sensitive to point-by-point errors that have a particularly strong influence on large images. Conversely, SSIM emphasizes mid-scale structure and changes nonmonotonously, because the optimized loss is point-by-point.

From Table 3, we also notice that our autoencoder is not overfitting, because after traning, $RMSE_{AE}$ for the training and the validation datasets are close, and testing $RMSE_{AE}$ is slightly lower. An interesting fact is that testing SSIM is much higher than that of the validation quality. This is one more confirmation of $RMSE_{AE}$ and SSIM noncorrelation.

The examples of autoencoder reconstructions are shown in Fig. 13.

Figure 13 shows that the reconstructed images from the testing dataset are smoothed and do not resolve the wave structure of the sea surface. The reconstructed amplitude of the back-scatter signal decreases monotonously with increasing distance from the radar antenna. The absence of a wave structure in the output image explains the low value of SSIM. Nevertheless, Figs. 13b and 13d have different scales of patterns. Hence, we suppose that the images, reconstructed by our autoencoder, potentially contain information about U_{10} speed, and, as a result, partial information about the wind wave structure.

4.2. ResNet-50 Training and Evaluation

The details of the pretrained modified ResNet-50 (Subsection 3.4) training and evaluation are shown in Table 4.

For the first 40 epochs with a nontrained convolutional layer, the quality metric RMSE remains practically constant both on the training and the validation datasets. Moreover, much higher validation RMSE testifies to the fact that a fully-convolutional layer learns inefficiently.

The start of the training of convolutional weights strongly reduces the RMSE value on the validation dataset from 2.348 m to 1.918 m at the end of training. The same quality metric on the training dataset remains close to 1.8 m. In general, we suppose that within this architecture, training reaches saturation and is unable to improve the regression quality. This point of view is also supported by the fact that the quality on the testing dataset is comparable to the validation dataset and worse than that on the training dataset.

Unsurprisingly, the baseline model of the modified ResNet-50 without pretrained convolutional part shows a better regression quality. RMSE without pretraining is assessed as 0.393 m on the testing dataset. This value is 5 times worse than that of

Table 4. Modified ResNet-50 training and evaluation

Epoch	$RMSE$, m	
	Training dataset	Validation dataset
Convolutional part is frozen		
2	1.787	2.341
20	1.780	2.355
40	1.786	2.348
Convolutional part is unfrozen		
60	1.834	1.935
100	1.827	1.941
140	1.842	1.918
	Testing dataset	
	1.927	

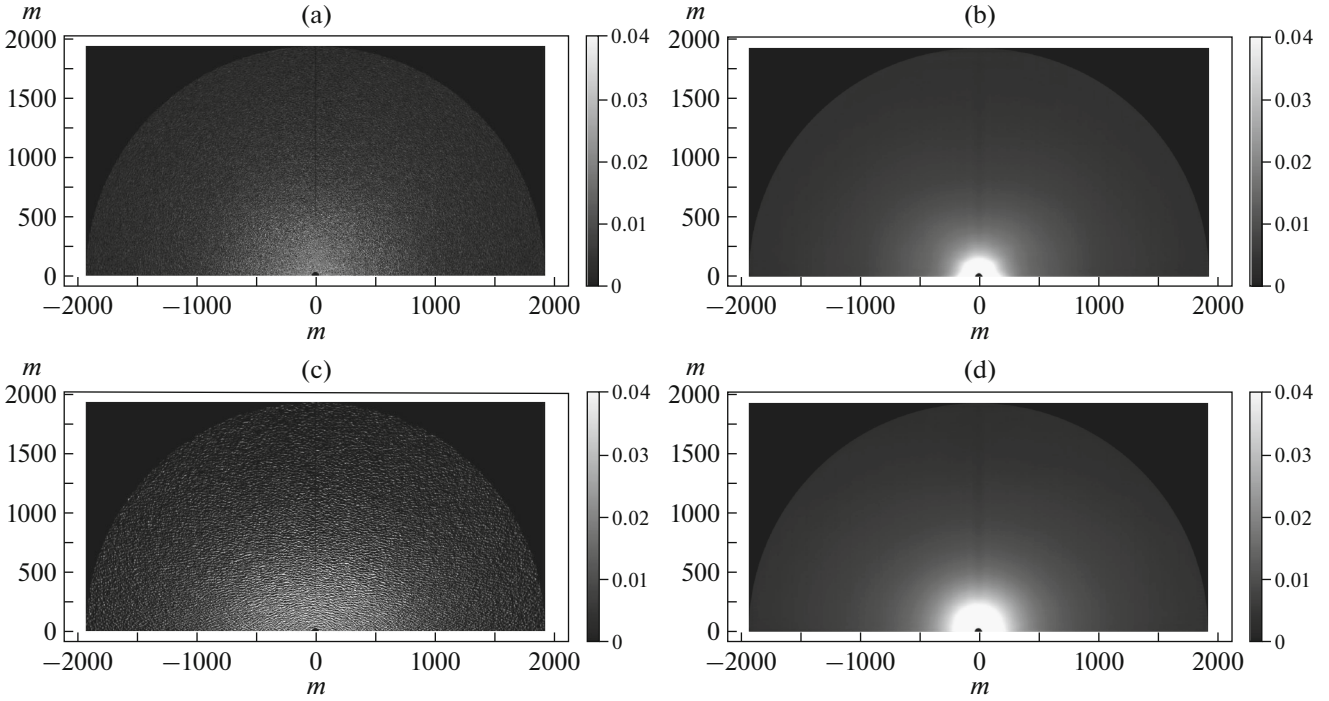

Fig. 13. Examples of autoencoder reconstruction of synthetic radar images with different U_{10} wind speeds: (a) 4 m/s input; (b) 4 m/s reconstructed ($RMSE_{AE} = 0.510$, $SSIM = 0.034$); (c) 12 m/s input; (d) 12 m/s reconstructed ($RMSE_{AE} = 0.689$, $SSIM = 0.042$).

the pretrained model (Table 4). We attribute this to the fact that although the autoencoder weights allow for extracting some features connected with SWH, they strongly distort the whole pattern of real radar images. The frozen convolutional part interferes with the extraction of the features of real radar images necessary for SWH determination. Thus, due to this distortion, the start of the training of the convolutional part optimizes parameters less efficiently than it does in the baseline model.

5. DISCUSSION

This study initially suggests that neural-network regression methods are effective for SWH determination from radar images. However, in situ data collection is expensive and is unable to provide diverse sea wave conditions.

The generation of synthetic radar images solves the problem of a dataset insufficiency. In addition, the generation of different synthetic sea surface conditions can simulate radar images with any hyperparameters necessary for the research. Synthetic images expand the dataset making it more or less uniform.

We propose a modified method of generating a synthetic sea surface and, consequently, synthetic radar images for the condition of a fully-developed sea. We consider this method to be promising because it enables the use of different power spectra of wind waves. The generated images are realistic for visual perception.

We make experiments of the direct pretraining of the autoencoder on a fully synthetic dataset. The convolutional neural network ResNet-50, modified for radar images, performs well in the task of SWH determination, especially, for a low wind speed. To expand generalization possibilities of this model, we propose to train this model on real data with the weights trained on a uniform synthetic dataset.

The results show that the autoencoder is unable to resolve the harmonical structure of wind waves. Based on it, we assume that MSE training loss, sensitive to point-by-point distortions, smoothes images and excludes small-scale features. On the contrary, it is the small-scale structure that contains essential information about wave lengths and heights. As a result, the pretrained ResNet-50, losing wave information, shows a worse regression quality than it was initially supposed to.

6. CONCLUSIONS

In this study, we presented the method for pretraining a convolutional artificial neural network based on ResNet blocks using the simulated radar

imagery of sea clutter generated under an assumption of stationary well-established wind-induced ocean waves. We presented the step-by-step algorithm for generating sea clutter radar images using forward and backward two-dimensional Fourier transforms under the assumption of the rough sea surface spectrum being the Pierson–Moskowitz spectrum parameterized by wind speed only. We demonstrated that the convolutional part of an artificial neural network performing the SWH regression task may be pretrained using simulated radar images. We then argue that the quality metrics of the pretrained network in the SWH regression task appears to be worse compared to the network trained from the scratch using only in situ target data for supervision.

We emphasize though that nonclassical contemporary methods of AI-based radar data processing still remain effective and convenient. This group of methods is radar-independent and easy to use. Yet these AI-based models require proper learning.

The results of our study suggest solving several problems regarding the proper use of synthetic sea clutter images for the pretraining approach to become effective. Therefore, the assessment of synthetic and semi-synthetic data for SWH estimation is still a topic of ongoing research.

We believe that the demonstrated performance of synthetic data generation should motivate further research towards the use of a variety of wave power spectra and the improvement of optical effects. One more direction of further research is searching for suitable pretraining models, capable of extracting useful features without loss of small-scale wave characteristics. We are strongly convinced that proper pretraining of a neural model may improve its performance in the SWH regression task.

FUNDING

A method for synthesizing sea clutter images has been developed with support from the program FMWE-2022-0002. The development and assessment of the artificial neural network was supported by the Kazan Federal University Strategic Academic Leadership Program (PRIORITY-2030).

CONFLICT OF INTEREST

The authors of this work declare that they have no conflicts of interest.

REFERENCES

1. K. Höhlein, M. Kern, T. Hewson, and R. Westermann, Meteorol. Appl. **27**, e1961 (2020). https://doi.org/10.1002/met.1961
2. S. Gulev, V. Grigorieva, and A. Sterl, Global Atlas of Ocean Waves at the Sea Atmosphere Interaction And Climate Laboratory (2014). http://www.sail.msk.ru/atlas/.
3. K. Hessner, J. Borge, and K. Reichert, in *Proc. 28th Int. Conf. Offshore Mechanics and Arctic Engineering* (ASME, 1999), p. 17.
4. N. Tilinina, D. Ivonin, A. Gavrikov, et al., Preprint ESSD-2021-431 (2021). https://doi.org/10.5194/essd-2021-431
5. N. Tilinina, D. Ivonin, A. Gavrikov, et al., Earth Syst. Sci. Data **14**, 3615 (2022). https://doi.org/10.5194/essd-14-3615-2022
6. R. Vicen-Bueno, C. Lido-Muela, J. Nieto-Borge, EURASIP J. Adv. Signal. Process. 84 (2012). https://doi.org/10.1186/1687-6180-2012-84
7. G. Ludeno and F. Serafino, J. Mar. Sci. Eng. **7**, 432 (2019). https://doi.org/10.3390/jmse7120432
8. G. Mastin, P. Watterberg, and J. Mareda, IEEE Comput. Graphics Appl. **7**(3), 16 (1987). https://doi.org/10.1109/MCG.1987.276961
9. W. Pierson and L. Moskowitz, J. Geophys. Res. **69**, 5181 (1964). https://doi.org/10.1029/JZ069i024p05181
10. D. Lyzenga and D. Walker, IEEE Geosci. Remote Sens. Lett. **12**(12), 2389 (2015). https://doi.org/10.1109/LGRS.2015.2478390
11. D. Hasselmann, M. Dunckel, and J. Ewing, J. Phys. Oceanogr. **10**, 1264 (1980).
12. J. Nieto Borge, G. Rodriguez, K. Hessner, and P. Gonzales, J. Atmos. Ocean. Technol. **21**, 1291 (2004). https://doi.org/10.1175/1520-0426(2004)021%3C1291:IOMRIF%3E2.0.CO;2
13. V. Rezvov, M. Krinitskiy, and S. Gulev, in *The 6th Int. Workshop on Deep Learning in Computational Physics* (2022), p. 429. https://doi.org/10.22323/1.429.0023
14. K. He, X. Zhang, Sh. Ren, and J. Sun, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition, Las Vegas, 2016* (IEEE, 2016), pp. 770–778. https://doi.org/10.1109/CVPR.2016.90
15. D. Misra, "Mish: A self regularized non-monotonic activation function," arXiv Preprint (2019). https://doi.org/10.48550/arXiv.1908.08681

16. E. Matsuyama, J. Biomed. Sci. Eng. **13**, 140 (2020). https://doi.org/10.4236/jbise.2020.137014

17. M. A. Kramer, AIChE J. **37**, 233 (1991). https://doi.org/10.1002/aic.690370209

18. T. A. Tani, M. M. A. Shibly, Md. Sh. Hasan, et al., in *Deep Learning Applications in Image Analysis*, Ed. by S. Roy, C.-H. Hsu, and V. Kagita, Studies in Big Data, Vol. 129 (Springer, Singapore, 2023), pp. 1–26. https://doi.org/10.1007/978-981-99-3784-4_1

19. W. Shi, J. Caballero, F. Huszár, et al., "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," arXiv Preprint (2016). https://doi.org/10.48550/arXiv.1609.05158

20. Y. Sugawara, S. Shiota, and H. Kiya, APSIPA Trans. Signal Inf. Process. **8**, e9 (2019). https://doi.org/10.1017/ATSIP.2019.2

21. P. Bergmann, S. Löwe, M. Fauser, et al., "Improving unsupervised defect segmentation by applying structural similarity to autoencoders," arXiv Preprint (2018). https://doi.org/10.48550/arXiv.1807.02011

22. D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," arXiv Preprint (2015). https://doi.org/10.48550/arXiv.1412.6980

Publisher's Note. Allerton Press remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.