




Article

The Classification of Noise-Afflicted Remotely Sensed Data Using Three Machine-Learning Techniques: Effect of Different Levels and Types of Noise on Accuracy

Sornkitja Boonprong ^{1,2} , Chunxiang Cao ^{2,*}, Wei Chen ² , Xiliang Ni ² , Min Xu ² and Bipin Kumar Acharya ^{1,2}

¹ University of the Chinese Academy of Sciences, Beijing 100094, China; sornkitja@radi.ac.cn (S.B.); bipin@radi.ac.cn (B.K.A.)

² State Key Laboratory of Remote Sensing Science, Institute of Remote Sensing and Digital Earth, Chinese Academy of Sciences, Beijing 100101, China; chenwei@radi.ac.cn (W.C.); nixl@radi.ac.cn (X.N.); xumin@radi.ac.cn (M.X.)

* Correspondence: caocx@radi.ac.cn; Tel.: +86-10-6480-6228

Received: 8 May 2018; Accepted: 26 June 2018; Published: 12 July 2018



Abstract: Remotely sensed data are often adversely affected by many types of noise, which influences the classification result. Supervised machine-learning (ML) classifiers such as random forest (RF), support vector machine (SVM), and back-propagation neural network (BPNN) are broadly reported to improve robustness against noise. However, only a few comparative studies that may help investigate this robustness have been reported. An important contribution, going beyond previous studies, is that we perform the analyses by employing the most well-known and broadly implemented packages of the three classifiers and control their settings to represent users' actual applications. This facilitates an understanding of the extent to which the noise types and levels in remotely sensed data impact classification accuracy using ML classifiers. By using those implementations, we classified the land cover data from a satellite image that was separately afflicted by seven-level zero-mean Gaussian, salt-pepper, and speckle noise. The modeling data and features were strictly controlled. Finally, we discussed how each noise type affects the accuracy obtained from each classifier and the robustness of the classifiers to noise in the data. This may enhance our understanding of the relationship between noises, the supervised ML classifiers, and remotely sensed data.

Keywords: machine learning; remote sensing; robustness; Gaussian noise; salt-pepper noise; multiplicative noise

1. Introduction

Remotely sensed data, especially for satellite images, are used to estimate information about the Earth, its various objects, phenomena, and processes. These images have been widely used for applications of Earth surface monitoring such as land cover classification and change detection, crop yield estimation, and geographic information extraction. Improving the accuracy of the classifications is thus a fundamental research topic in the field of geographic information sciences [1]. However, it can be a difficult task depending on the complexity of the landscape, the spatial and spectral resolution of the imagery being used, and the amount of noise included.

Noise may be added to remotely sensed data during different stages of data acquisition or processing, from the moment the data are captured by the sensor until their atmospheric and topographic correction, orthorectification, or co-registration [2]. In satellite image classification,

the data obtained from the satellite sensors may be affected by atmospheric noise resulting from the obstruction of the light reflected by targets on the Earth's surface by a variety of phenomena, including aerosols and clouds in the atmosphere as well as changing illumination patterns and the angle at which the satellite views the ground at any given time [3]. The basic source for image pixels, such as those acquired by satellite sensors, has limited radiometric and geometric resolution. This effect leads to a mixture of classes within one pixel, thereby resulting in the representation of the partial degree membership of at least two classes in the pixel, known as a mixed pixel, which normally increases the classification complexity [4]. Moreover, the image generation process may add noise to the data. This is widely found in most Synthetic-aperture radar (SAR) images [5–7]. These data have to be compressed to reduce their requirements for archiving and data transmission and this may cause artifacts and ambiguities in the final images [8].

Normally, the normal distribution values that occur at a random location by merging with the original values in the satellite image are referred to as zero-mean Gaussian noise [9]. In optical remote sensing multispectral imagery, the noise is typically independent of the data and it is generally additive in nature. The noise may reduce the performances of important techniques of image processing such as detection, segmentation, and classification. Most of the natural images are assumed to have additive white Gaussian noise [10,11].

Salt and pepper noise are the highest and lowest global values, respectively, that replace an original pixel at a random location. This is generally caused due to errors in data transmission [5,12]. A report of data loss on the Landsat Missions website also refers to an artifact called Christmas Tree anomaly. The lost data appear as bright red, green, and blue artifacts, often next to or overlapping each other. This artifact is usually caused by telemetry data erroneously being included in the satellite imagery [13]. The missing data may be replaced by null values (salt noise) or filled with a designated fill pattern by the ground processing systems. Charoenjit et al. [14] stated that one of the problems associated with biomass estimation from very high-resolution (VHR) data is salt–pepper noise. In an attempt to solve the problem caused by the noise, these authors performed object-based image analysis on a Para rubber plantation and proved that the segmentation method can reduce the effect of constant salt–pepper noise.

Multiplicative noise or speckle is widely found in most SAR images [5–7]. These data have to be compressed to reduce their requirements for archiving and data transmission and this may cause artifacts and ambiguities in the final images [8,15]. Speckle noise gives a grainy appearance to radar imageries. It reduces the image contrast, which has a direct negative effect on the texture-based analysis of the imageries [16,17]. For example, Nobrega et al. [18] showed a LiDAR intensity image that was affected by signal eccentricities caused by sensor scanning patterns. By filtering the noise, they could improve the efficiency of the image's segmentation. Moreover, Mulder et al. [19] reviewed the use of remotely sensed data for soil and terrain mapping. They concluded that the accuracy of retrieving soil information by using proximal sensing combined with a remote sensing method decreases because of several types of noise such as that caused by a mixture of soil properties and atmospheric, topographic, and sensor noise. Soil moisture retrieval from SAR images is always affected by speckle noise and uncertainties associated with soil parameters, which impact negatively the accuracy of soil moisture estimates. Barber et al. [7] proposed a soil moisture Bayesian estimator from polarimetric SAR images to address these issues. The results indicated that the model enlarges the validity region of the minimization-based procedure and the speckle effects can be reduced by the multilooking method included in the model scheme.

Consequently, the effect of noise within the data can be either mitigated or ignored by selecting appropriate classifiers that are robust to noise, especially for supervised machine-learning classifiers which have a mechanism for handling noise [20–23]. Continuous developments in storage capacity and the processing speed of computers made advanced machine learning (ML) available for land cover classification. Supervised machine-learning algorithms require external assistance in the form of training [24]. Their classification accuracy obviously depends on the quality of the training data [20].

The input dataset is normally divided into training and testing datasets. The training dataset has an output variable that needs to be predicted or classified. Most algorithms learn some kind of pattern from the training dataset and apply them to the test dataset for prediction or classification [25,26].

Algorithms based on random forests (RF) [21] utilize decision trees. These algorithms are probably one of the most efficient ML algorithms in terms of prediction accuracy [27]. The algorithm has the capability to rapidly process databases ranging from small to very large, and is easy to interpret and visualize [28]. Moreover, RF has been reported to be robust to noise [29–31]. However, the algorithm can slow down as the number of trees increases [28]. In recent decades, many scientists have used RF as a classifier. In the original paper describing RF [21], the performance of RF was observed against Adaboost algorithms [32] by classifying 20 different datasets. The authors proved that RF is more robust with respect to noise. Moreover, Crisci et al. [27] reviewed a number of supervised ML algorithms to model mortality events in benthic communities in rocky coastal areas in the northwestern Mediterranean Sea. The RF model yielded the lowest misclassification rates for such ecological data.

Support vector machine (SVM) is a supervised non-parametric statistical learning technique; therefore, no assumption is made about the underlying data distribution [33]. The SVM learning algorithm aims to find a hyperplane that separates the dataset into a discrete predefined number of classes in a fashion consistent with the training examples. SVM outperforms other ML models, particularly when only a small dataset is available for training [34,35]. However, SVM requires more training time and its performance is dependent on parameter adjustment in comparison to other methods [23]. For classification purposes, Dalponte et al. [36] used SVM and the fusion of hyperspectral and LiDAR data, in which speckle noise usually existed. They pointed out that SVM outperformed the maximum likelihood and k-nearest neighbor (k-NN) techniques. Moreover, the incorporation of LiDAR variables generally improved the classification performance and the first return data was the highest contributing factor. Insom et al. [34] improved the accuracy of standard SVM by applying a particle filter (PF) to automatically update the SVM training model parameters to values that were more appropriate for their flood dataset. The performance of the method is superior when compared with standard SVM. Senf et al. [37] performed land cover classification of complex Mediterranean landscapes using an SVM classifier. The work proved that combining phonological profiles of the land covers and human interpretation can significantly improve the generalization power of the SVM models, as previously stated [38]. Moreover, SVM was also reported to have good generalization capabilities under different noise levels, types of noise, and sample sizes when it was optimized [39].

A neural network (NN) is an algorithm that simulates the neuronal structure, processing method, and learning ability of the human brain but on much smaller scales. This technique is applicable to problems in which the relationships may be nonlinear or quite dynamic [40]. The most common type of NN is based on the back-propagation learning algorithm and is known as a back-propagation neural network (BPNN) [41]. BPNN has been proven to be the best among the Multi-layer perceptron (MLP) algorithms [22]. However, BPNN tends to be slower to train than other types of NNs, which can be problematic in very large networks with a large amount of data [28]. BPNNs in remotely sensed image classification applications have been widely reviewed [41–43]. They were also reported as very effective to use in noise reduction [44,45] and robust to noises when trained by noise data [46]. These researchers have mutually concluded that the BPNN approach is feasible for the classification of remote sensing imagery.

In recent decades, many experiments have been designed for studies aiming to observe the robustness of a variety of classifiers relating to noise content and type. For example, Petrou et al. [2] proposed fuzzy-rule-based classifiers to overcome uncertainty in habitat mapping. The robustness of these classifiers was evaluated by drawing additive homogeneous noise from a zero-mean Gaussian probability density function. The noise was added to the original images and a threshold was assigned by an expert at different standard deviations ranging from 5 to 20% of the image mean values and 5 to 50% of the rule threshold, respectively. The results showed that the classifiers remained mostly unaffected by noise in the data that were used because the classification was object based. In contrast,

the additive noise in the rule threshold, which acted as inaccurate expert rules, directly affected the performance. However, the overall accuracies remained over 75%. Celik and Ma [9] investigated the performance of their automatic change-detection method by using different types and levels of added noise. Different levels of zero-mean Gaussian noise ranging from 35 to 50 dB (peak signal-to-noise ratio (PSNR)) were used in the experiments. Moreover, the authors contaminated the test images with different levels of speckle noise to illustrate the noise that commonly occurs in SAR data. The method proved to be highly robust (accuracies of approximately 95–100%) to both types of noise when the classifier was assigned an appropriate parameter value. Moreover, the noise test could assist the authors in optimizing the parameter used in the classification with different levels of noise. Dierking and Dall [6] used complex scattering matrix data of the Electromagnetics Institute's Synthetic Aperture Radar (EMISAR) system in order to simulate image products with different levels of pixel dimensions and noise levels of -35 , -30 , -25 , and -20 dB. Ice deformation maps were generated by using a selected threshold, such that image pixels with intensities equal to or larger than the highest 2% of the ice intensity distribution level were classified as deformed ice. The results indicated that if the level of the sensor noise is equal to or larger than the average backscattering intensity of the level ice, the retrieved values for the deformation parameters change. Moreover, an L-band SAR system at like-polarization, with an incidence angle larger than 35° and a noise level of at most -25 dB or lower, is desirable for mapping the deformation state of sea ice cover.

In this work, we employed three supervised ML classifiers, i.e., random forest (RF), support vector machine (SVM), and back-propagation neural network (BPNN), to classify the land covers in satellite images afflicted by three classes of noise, namely zero-mean Gaussian, salt-pepper, and speckle noise. The classification was first carried out on an image corrected for absolute atmospheric interference, termed the reference image, to obtain the standard accuracies. Subsequently, noise was applied to the reference image in different concentrations. The normal reflectance image (without absolute atmospheric correction) was also classified to illustrate the real effects of mixing different kinds of noise. To conclude, we experimented with the performance of the classifiers under the following noise conditions:

1. Fully processed images with absolute atmospheric correction, termed the reference images. (assumed to contain no noise);
2. Normal reflectance images without absolute atmospheric correction (mixture of noise, i.e., sensor noise, atmospheric noise);
3. Reference images with zero-mean Gaussian noise added;
4. Reference images with salt-pepper noise added;
5. Reference images with speckle noise added (multiplicative noise).

Discussions are presented to describe and examine the performance of the different classifiers to each noise type and level in addition to assessing the research questions: (1) To what extent do the different types of noise affect the classification accuracies? (2) How do the classifiers behave toward noise-afflicted image classification? Additionally, the effect of the different types of noise and the behavior of the classifiers on remotely sensed data are considered. Note that the results and conclusions that we made in this study are mainly aimed at providing an understanding of the core mechanisms of the ML classifiers toward noises in remotely sensed data rather than comparing their performances.

2. Materials and Methods

2.1. Experimental Design

Figure 1 shows the overall experimental flow of the current study. A Landsat 8 OLI scene was acquired and processed to be a reference image (refer to Section 2.2. for the image processing). We added different levels of three types of noise, zero-mean Gaussian noise, salt-pepper noise, and speckle noise to the reference image to produce noise-afflicted images. Twenty-three images (one

reference image, one image without absolute atmospheric correction, and 21 images to which artificial noise was added) were then classified by BPNN, SVM, and RF to obtain the classification accuracies. The classifications were scoped by the following conditions. First, the training and validating samples were extracted using the same shape file (i.e., samples were acquired at the same image locations). Second, the parameters used in each of the algorithms were optimized. Please refer to Section 2.4 for a detailed explanation. In addition, we observed the robustness of a classifier by the differential accuracies of the reference image and the images to which noise was added.

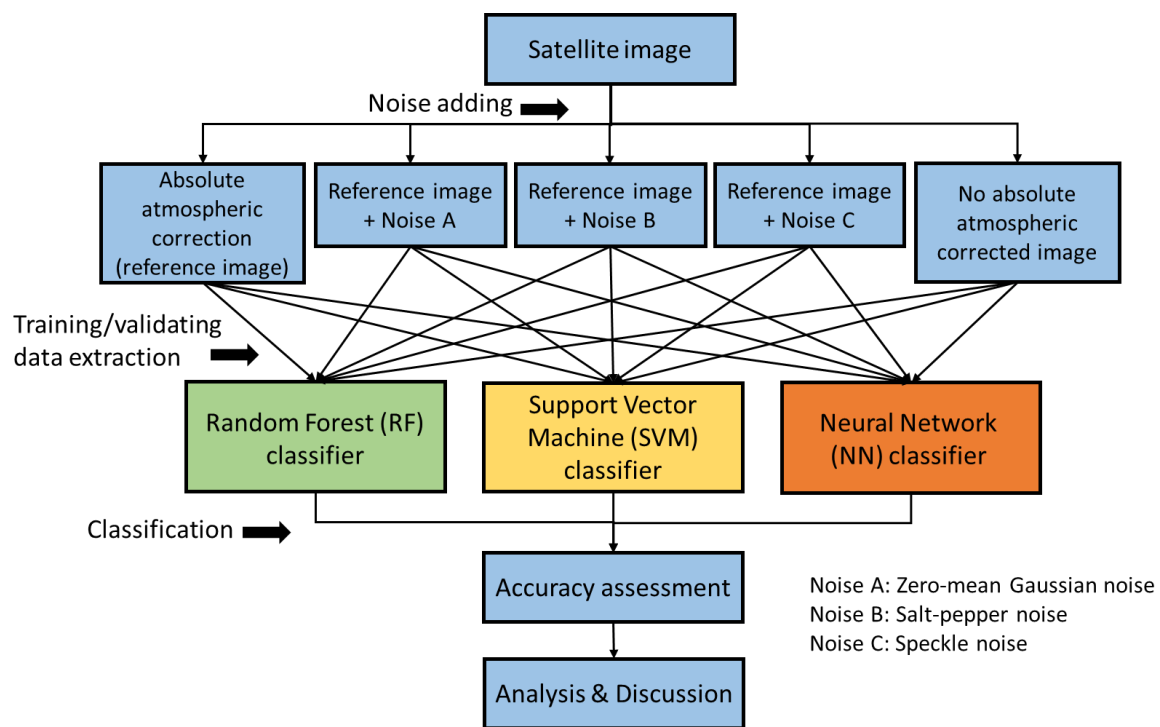


Figure 1. Experimental flow.

2.2. Remotely Sensed Data

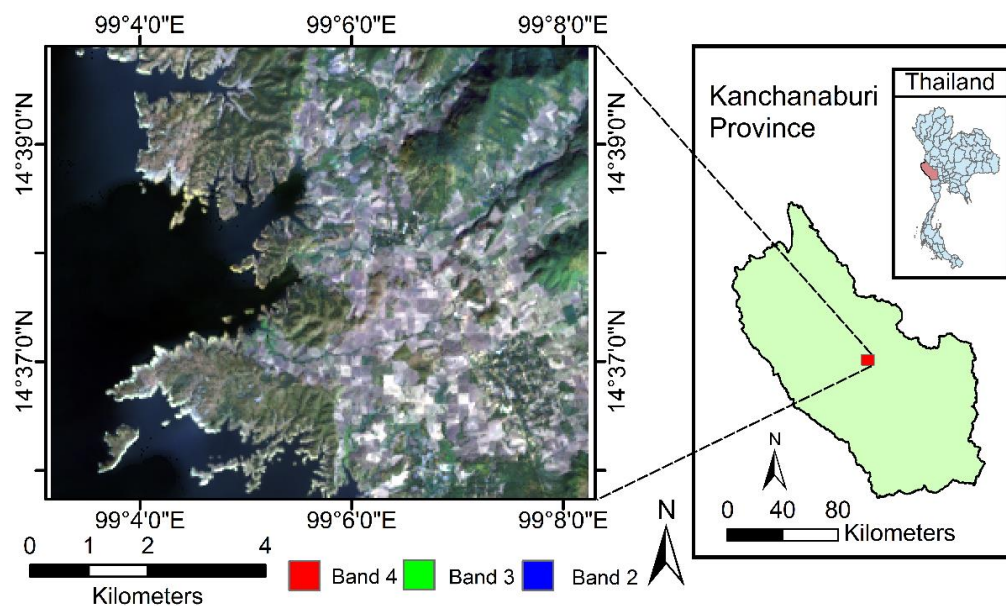
The satellite image is a Landsat-8 Operational Land Imager (OLI) image acquired on 26 December 2014 (Path 130, Row 50), shown in Figure 2. The image was acquired under clear sky conditions, a very low content of homogeneous aerosol, and in the absence of cloud objects. The scene includes Landsat 8 surface reflectance data generated from the Landsat Surface Reflectance Code (LaSRC). The LaSRC makes use of the coastal aerosol band to perform aerosol inversion tests, uses auxiliary climate data from the Moderate Resolution Imaging Spectroradiometer (MODIS), and uses a unique radiative transfer model. The model artifacts or blockiness presented in the surface reflectance data products were excluded from the image and any future calculation by employing the quality assessment (QA) band. We termed this image the reference image.

The study area, which covered the southeastern part of the Srinakarin dam, an embankment dam on the Khwae Yai River in the Si Sawat District of the Kanchanaburi Province of Thailand, included five land cover classes: agriculture, bare land, construction, water body, and forest. Descriptions of the classes are provided in Table 1. The sample pixel locations were randomly selected from Thailand's 2014 land cover ground survey map obtained from the Geo-Informatics and Space Technology Development Agency (GISTDA). The phenology of plants or land cover may change after the referencing date (the time lapse is about two months). Thus, we incorporated human interpretation and Google Earth images with an acquisition date near the Landsat image to confirm that the samples had not changed category during the time lapse.

Table 1. Number of samples of modeling and application testing datasets.

Classes	Descriptions	M *	A *
Agriculture	Artificial planting area/no harvested area included	504	126
Bare land	Land used without vegetation cover	440	110
Construction	Houses, roads, and any man-made construction	80	20
Water body	Water bodies such as ponds, lakes, and rivers	344	86
Forest	Area covered by natural and unused vegetation, mountain shadow	2048	512
Total		3416	854

* Modeling data (M) and application testing (A) data.

**Figure 2.** True color image of the study area.

2.3. Sampling Design

The sampling framework utilized in this paper can be explained as follows. (1) Four thousand and four hundred random samples were randomly generated based on the Thailand's 2014 land cover ground survey map via *Generate Random Sample—Using Ground Truth Classification Image* in ENVI classic version 5.1. (Harris Geospatial Solutions, Inc., Melbourne, Australia) The random samples include all classes except the construction class because it rarely existed in the study area; therefore, we manually selected 100 samples of the construction class by human interpretation. The initial 4550 random samples include 2600 samples of the forest class, 650 samples of the agriculture class, 650 samples of the bare land class, and 650 samples of the water body class. Note that the forest class accounts for more than 50% land cover of the study area, so we provided a sample size three times larger than other classes to capture the forest diversity. On the other hand, we maintained the same proportion for the water body class as it has less complexity. As a result, 4650 samples were obtained in this step. (2) The samples close to the edge of the class vectors or their classes changed during the time lapse of the data acquisition and survey map and were excluded. The criterion for removing the edge's pixels is that the distance from the vector line is less than three pixels or 90 m. The remaining pixels were compared one-by-one with the Google Earth images using human interpretation to detect any category changes. (3) Finally, 4270 samples remained in the experiment. An 80/20 ratio of training and validation samples was randomly drawn without replacement in MATLAB programming. The final sample distribution is shown in Table 1.

To ensure that the accuracy of the classifiers was comparable and less biased, 80% of data points was used to train the various classification algorithms and 20% was used for comparison. This means that all of the training elements required for generating a classification model—training set, testing set, or validating set (if needed)—were included in the 80% training data. Note that each classifier used this data in different ways based on the training algorithms. The management of the training data of each classifier can be found in Section 2.5. Finally, we distinctly preserved 854 validating samples that were sparse in the image (20% of the reference data) to evaluate the classification accuracies. Note that only the Coastal, Blue, Green, Red, Near-Infrared (NIR), Shortwave-Infrared 1 and 2 (SWIR1 and SWIR2, respectively) bands of the Landsat 8 OLI dataset were used as spectral features in this study.

2.4. Noise Afflictions

We investigated the highest case of noise affliction in the satellite image by simulating the types of noise mentioned below in the MATLAB environment, which was then applied to the reference image at different levels of noise. The Gaussian distribution noise can be expressed by:

$$P(x) = 1/(\sigma\sqrt{2\pi}) \times e^{-\frac{(x-\mu)^2}{2\sigma^2}}, \quad (1)$$

where $P(x)$ is the Gaussian distribution noise in an image; μ and σ are the mean and standard deviation of the noise process, respectively.

Salt and pepper noise are the highest and lowest global values, respectively, that replace an original pixel at a random location. To simulate salt and pepper noise, the following conditions were applied to the noise-free image:

$$p(x) = \begin{cases} p1, x = A \\ p2, x = B \\ 0, otherwise \end{cases}, \quad (2)$$

where $p1$ and $p2$ are the probabilities density function (pdf); $p(x)$ is the distribution of salt and pepper noise in image; A, B are the possible minimum and maximum values, respectively (in this paper they are 0 and 1, respectively).

The speckle noise is known as multiplicative noise, i.e., it is produced by the coherent superposition of spatially random multiple scattering sources within the resolution volume of the sensor [47]. The speckle noise distribution can be expressed as:

$$J = I + n \times I, \quad (3)$$

where J is the distribution speckle noise image; I is the input image; and n is the uniform noise image by specific mean and variance.

The noise level is quantitatively defined in decibels (dB) in terms of the PSNR. Given an input image I and its noisy image H , the PSNR between the two images can be expressed as:

$$\text{MSE} = (1/mn) \sum_{i=1}^m \sum_{j=1}^n [I(i,j) - H(i,j)]^2, \quad (4)$$

where i and j are pixel locations in rows and columns, and m and n are the maximum number of image rows and columns, respectively.

Note that a smaller value of the PSNR refers to increasing noise intensity. The noisy images were produced by using different PSNR values. Finally, the sample images used in this study comprise a total of 23 datasets with the following descriptions:

Reference images (applied absolute atmospheric correction and noise removal)

Level 1 Landsat 8 images without absolute atmospheric correction (only convert from digital number to reflectance values)

Reference images + zero-mean Gaussian noise from 10 to 40 dB (in steps of 5 dB)

Reference images + salt and pepper noise from 10 to 40 dB (in steps of 5 dB)

Reference images + speckle noise from 10 to 40 dB (in steps of 5 dB)

Note that each pixel value of the reference image was rescaled to values between 0 and 1 using the minimum and maximum values (as shown in Equation (5)) of its band before noise generation. If an image data has i rows, v columns, and u bands, the rescaling equation in this work is expressed as:

$$I(i, v, u) = (Q(i, v, u) - \min(u)) / (\max(u) - \min(u)), \quad (5)$$

where $I(i, v, u)$ is the rescaled value of the original value $Q(i, v, u)$. The terms $\min(u)$ and $\max(u)$ denote the minimum and maximum values selected from band u , respectively.

2.5. Classifiers—Implementation Packages

The classifiers are described in terms of their key concepts, operational details, and parameter settings that are exclusively used in this study. Further details of both the theoretical and mathematical descriptions can be found by linking to the appropriate work that provides extensive points of interest (RF: [21,48,49], SVM: [50,51], BPNN: [22,42]).

2.5.1. Random Forests (RF)

Random Forests build multiple decision trees based on random bootstrapped samples of the training data. In contrast to other classifiers, RF neither causes overfitting nor does it require a long training time [21]. Each tree is built using a subset that differs from the original training data, containing approximately two-thirds of the cases, and the nodes are split using the best split variable among a subset of m randomly selected variables [52]. The trees are created by drawing a subset of training samples by using the replacement or bagging method. Each decision tree is independently produced without any pruning and each node is split using a number (m) defined by the user. By growing the forest up to the number of trees (k), the algorithm creates trees with high variance and low bias [21]. The final classification decision is taken by averaging the class assignment probabilities calculated by all trees. Therefore, two parameters are required to construct an RF framework: the number of trees ($ntree$) in the ensemble and the number of variables used to split the nodes ($mtry$). The classifications were performed using the well-known “randomForest” package for R [43].

All of the considered parameters were investigated, namely, the $ntree$ was observed from 500 to 2000, and the $mtry$ was observed from 2 to 5. The resulting model was selected by choosing the most accurate model that was obtained by running the model generation procedure 30 times. Table 2 shows the parameter tuning of all classifiers in this study.

Table 2. Parameter tuning of all classifiers.

Classifiers	Cross-Validation Approaches/Parameters
Random forests (RF)	$ntree = \{500, 1000, 1500, 2000\}$, $mtry = \{2, 3, 4, 5\}$, 30 replications
Support vector machines (SVM)	Grid-search cross-validation approach, radial basis function kernel, 30 replications
Back-propagation neural networks (BPNN)	Scale conjugated gradient optimization, 70% training, 15% testing, 15% validating, hidden nodes = $\{5, 10, 15, 20, 40, 60\}$, 30 replications

The optimized models are different for each observed image.

2.5.2. Support Vector Machines (SVM)

The SVM algorithm aims to find a decision boundary termed a hyperplane that separates the dataset into a discrete predefined number of classes in a fashion consistent with the training examples. In this study, we employed the well-known “LIBSVM” package [53] implemented in the MATLAB environment. A brief mathematic description of the SVM package is provided below.

Given the input training sets of l examples x_i with labels $y_i: (x_i, y_i), i = 1, 2, 3, \dots, l$ where $x_i \in R^N$ and $y \in \{1, -1\}^l$, the SVMs require the solution of the following optimization problem [53] as revised from The Soft Margin Hyperplane [51]:

$$\begin{aligned} & \max_{w, b, \xi} \frac{1}{2} w^T w + C \sum_{i=1}^l \xi_i, \\ & \text{subject to } y_i (w^T \phi(x_i) + b) \geq 1 - \xi_i, \xi_i \geq 0, \end{aligned} \quad (6)$$

where w, b, T , and ξ are the l -dimensional vectors, bias, transpose function, and non-negative variables, respectively. The training vectors x_i are mapped into a higher dimensional space by the function ϕ . SVM finds a linear separating hyperplane with the maximal margin in this higher dimensional space. $C > 0$ denotes the penalty parameters of the error term. Furthermore, $K(x_i, x_j) \equiv \phi(x_i)^T \phi(x_j)$ is termed the kernel function. In this work, the radial basis function (RBF) kernel is used. It is expressed as:

$$K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2), \gamma > 0, \quad (7)$$

where γ is the kernel parameter. Therefore, only two parameters—the penalty parameter C and kernel parameter γ —are required when using the method.

To attain reasonable accuracy in each of the classifications, we implemented the algorithm in a step-by-step manner. First, the training data were converted to a suitable form for the SVM package. The labels were changed from category attributes into binary numbers 0 and 1. For example, {water body, constructions, agriculture} were represented as (1,0,0), (0,1,0), and (0,0,1). The straightforward grid-search cross-validation approach [53], which is provided in the package, was performed to find the best model parameters, since the values of C and γ directly affect the performance of the SVM model [34]. The cross-validation was replicated 30 times (Table 2). Therefore, the final SVM model was generated by training the entire training set by the obtained parameters in regards to the testing set, after which we exported the model to classify the 20% validating data.

2.5.3. Back-Propagation Neural Network (BPNN)

The goal of BPNN is to compute the partial derivative or gradient of a loss function with respect to any weight in the network. The loss function calculates the difference between the input training example and its expected output, after the example has been propagated through the network [36]. The basic concern of an NN algorithm is that it cannot perform accurately beyond the range of trained inputs; that is, the classification accuracies strongly depend on the data used to train the networks. The modeling data were labeled by binary number forms and then they were randomly separated into training, testing, and validating sets in the ratios of 70%, 15%, and 15%, respectively. The number of hidden layers was observed from 5 to 60. The BPNN model was automatically trained until the network converges by using the `trainscg` function (scaled conjugate gradients or SCG), the details of which have been described by Moller [49], in the MATLAB environment. The SCG training was automatically optimized by the parameter sigma σ (which determines the change in weight for the second derivative approximation) and lamda λ (which regulates the indefiniteness of the Hessian). The values of σ and λ were taken as $5 \times e^{-5}$ and $5 \times e^{-7}$, respectively. The model was repeated 30 times and the best model was then employed to classify the given validating data. The mean and standard deviation of the model accuracies were reported.

3. Results

3.1. Image with Added Noise

Each noisy image was constructed by considering the PSNR values. An automatic noise-adding function was used to apply random noise to each satellite image band. In the case of salt-pepper noise, the pixel location may possibly be disturbed by the noise (irrespective of whether it is salt or

pepper) more than once, i.e., the first location (row 1, column 1) of the seven-band satellite image data consists of seven pixels, and the noise can either be added or not added to a pixel, some pixels, or all pixels. In the case of additive (zero-mean Gaussian noise) and multiplicative noise (speckle noise), all pixels in each image band were directly afflicted by adding or multiplying by random values. Note that, to produce the intended PSNR, a certain noise parameter is associated with each image band depending on the type of noise. In other words, a noise layer was prepared by using specific parameters such as the variance, mean, and standard deviation, after which the PSNR was calculated from the mean-square error of the original image and the noise layer combined with the original image. This requires the parameters to be assigned a certain value before they are used to target a certain PSNR. The noisy images are shown in Figure 3.

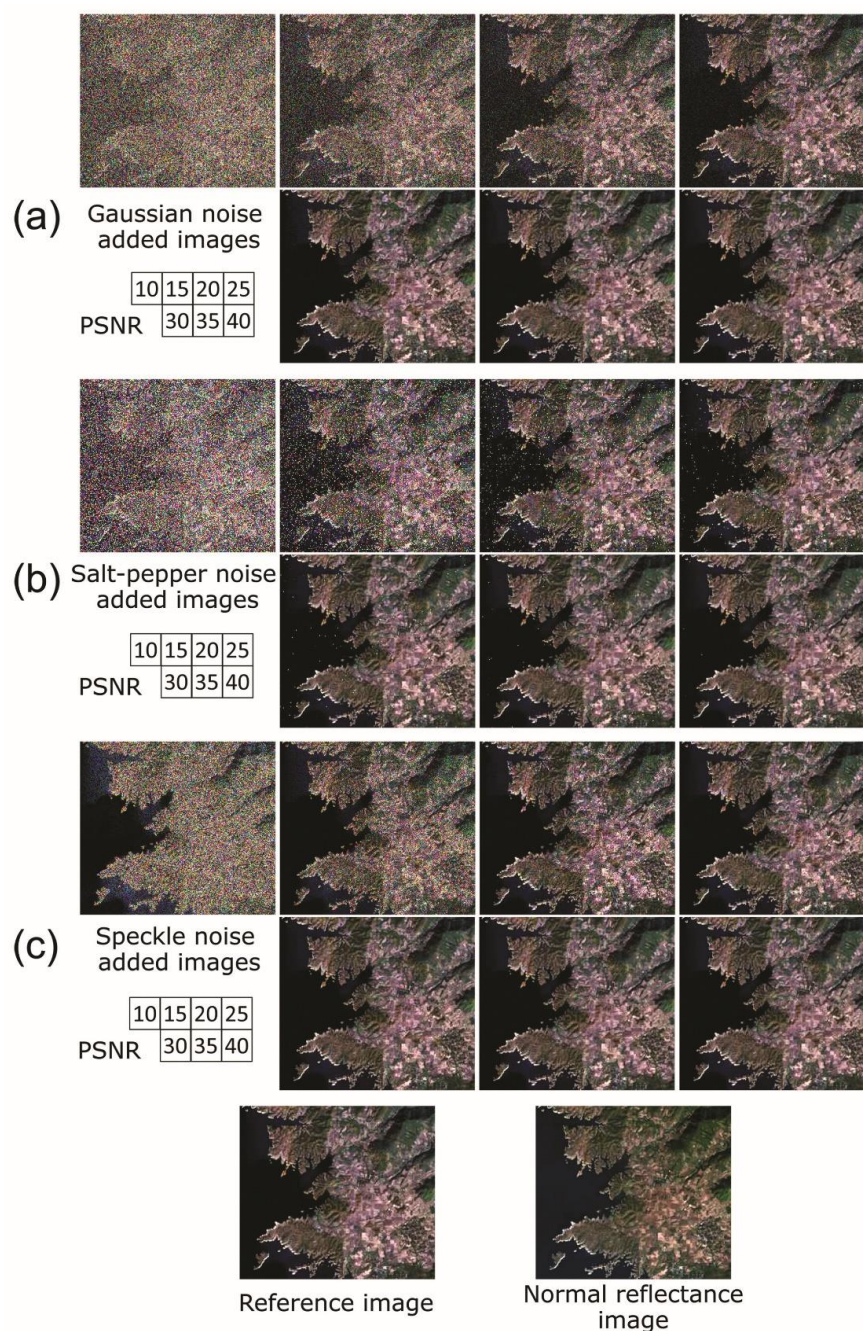


Figure 3. Twenty-three satellite images with added noise in different peak signal-to-noise ratios (PSNRs): (a) Gaussian noise, (b) salt-pepper noise, and (c) speckle noise (true color image; R = band 4, G = band 3, and B = band 2).

3.2. Classification Results

3.2.1. BPNN

The mean model accuracy (OA_{m1}) and mean validation accuracy (OA_{V1}) obtained by classifying (30 replications) the reference image by the neural network model are 99.1% and 96.7% ($K = 0.94$). In the case of images without atmospheric correction, the algorithm produced accuracies of 99.0% and 96.4% ($K = 0.94$) for OA_{m1} and OA_{V1} , respectively. It is obvious that the value of OA_V is fairly low when the algorithm encounters very high noise content. The best validation accuracy (OA_{V0}) of the images to which zero-mean Gaussian noise was added is 65.9% ($K = 0.31$) at 10 dB, for which the classifier was completely unable to classify the construction class until reaching a PSNR of 30 dB. The accuracy rapidly increased from this percentage to over 90% at 25 dB, then slightly increased to 98.5% ($K = 0.97$) at 40 dB.

The OA_{V0} for image classification with added speckle noise at 10 dB is 73.8% ($K = 0.52$), which differs considerably from the lowest accuracy obtained for the zero-mean Gaussian noise; however, the construction class still cannot be classified until a PSNR of 30 dB is reached. Above 25 dB, the image classification accuracies are higher for images to which zero-mean Gaussian noise was added in comparison to speckle noise. In contrast, the classification accuracies obtained for images with added salt–pepper noise are higher than those to which other types of noise were added, i.e., the mean validation accuracies (OA_{V1}) exceed 85% ($K \geq 0.75$) at 10 dB and are over 90% ($K \geq 0.85$) for the remaining PSNR levels. The classification accuracies obviously showed that BPNN is affected by all types of noise at increasing noise intensities. Details of the classification accuracies of the BPNN classifiers are presented in Table 3.

3.2.2. SVM

The mean model accuracy (OA_{m1}) and mean validation accuracy (OA_{V1}) obtained from classifying the reference image by the SVM model are as high as the values obtained from non-atmospheric-corrected images, i.e., approximately 99% ($K \geq 0.98$). The OA_{V0} of the images to which zero-mean Gaussian noise was added is 64.5% ($K = 0.23$) at 10 dB, for which the classifier was completely incapable of classifying the agriculture and construction class until PSNRs of 15 dB and 25 dB, respectively, were reached. In most of the cases, the accuracy increases from this percentage to 99.2% ($K = 0.99$) at 40 dB. The OA_{V1} for the classification of images to which speckle noise was added at 10 dB is 74.5% ($Kappa = 0.54$), which differs by 10% from the value obtained for zero-mean Gaussian noise; however, it is still not possible to classify the construction class until a PSNR of 30 dB is reached. At high noise content (10–20 dB), the classifications of images to which speckle noise was added yielded accuracies higher than those of the datasets afflicted by zero-mean Gaussian noise. Contrary to this, the mean validation accuracies are over 90% ($K \geq 0.83$) for all classification tasks that were afflicted by salt–pepper noise. Moreover, the SVM algorithm could model all classes to which this noise was added. Details of the classification accuracies of the SVM classifiers are provided in Table 4.

3.2.3. RF

In the case of RF classifiers, the mean model accuracy (OA_{m1}) and mean validation accuracy (OA_{V1}) obtained by classifying the reference image are 99.1% and 98.8% ($K = 0.98$), respectively. The algorithm produced the closed OA_{V1} value for the reference and image without atmospheric correction, which is consistent with the results obtained from the BPNN and SVM. The increasing trends observed for OA_{m1} and OA_{V1} for the classification of images to which zero-mean Gaussian and speckle noise was added is similar to that of SVM and BPNN, but the RF classifiers are more effective at high noise intensity. Unexpectedly, the mean validation accuracies reached 93.8% ($K \geq 0.89$) for 10 dB salt–pepper noise-afflicted image classifications. Consequently, classifying the construction class, the classification of which is problematic with both BPNN and SVM, remains impossible until PSNRs of 25, 15, and 30 dB are reached with zero-mean Gaussian, salt–pepper, and speckle noise, respectively. Details of the classification accuracies of the RF classifiers are provided in Table 5.

Table 3. Classification accuracies of the BPNN classifiers.

	RI	NI	G10	G15	G20	G25	G30	G35	G40	SP10	SP15	SP20	SP25	SP30	SP35	SP40	SPK10	SPK15	SPK20	SPK25	SPK30	SPK35	SPK40
PA1	99.2	97.6	7.1	38.1	69.8	87.3	92.1	91.3	96.0	78.6	94.4	90.5	95.2	96.8	98.4	96.8	27.8	46.8	72.2	84.9	94.4	91.3	93.7
PA2	97.3	99.1	46.4	81.8	86.4	95.5	97.3	97.3	98.2	90.9	96.4	97.3	97.3	98.2	99.1	100	44.5	76.4	90.0	95.5	97.3	97.3	99.1
PA3	95.0	85.0	0	0	0	0	5.0	70.0	85.0	0	0	0	75.0	80.0	95.0	95.0	0	0	0	0	30.0	65.0	80.0
PA4	100	100	39.5	76.7	91.9	100	100	100	100	81.4	91.9	100	97.7	100	100	100	87.2	100	100	100	100	100	100
PA5	99.8	99.6	91.6	91.8	94.3	96.9	98.0	99.0	99.4	97.7	97.5	97.3	97.9	99.6	99.4	99.6	92.0	93.9	94.3	96.3	97.7	98.8	99.8
UA1	98.4	99.2	47.4	54.5	71.5	85.3	89.9	98.3	98.4	88.4	88.8	92.7	92.3	98.4	99.2	100	40.7	56.7	73.4	82.3	89.5	95.0	99.2
UA2	98.2	95.6	57.3	81.1	87.2	90.5	91.5	93.9	94.7	84.7	92.2	90.7	94.7	96.4	98.2	98.2	53.8	75.0	84.6	87.5	93.9	93.0	97.3
UA3	95.0	100	0	0	0	0	50.0	87.5	100	0	0	0	83.3	88.9	90.5	90.5	0	0	0	0	75.0	92.9	94.1
UA4	100	100	60.7	77.6	94.0	96.6	100	100	100	94.6	98.8	94.5	100	100	100	100	87.2	93.5	100	100	100	100	100
UA5	99.8	99.4	68.0	82.5	89.8	95.4	96.5	97.3	99.0	91.2	95.0	95.6	98.4	99.2	99.6	99.4	79.7	88.1	91.7	95.2	97.5	97.7	98.3
OA _{m0}	99.7	99.9	90.6	93.7	96.4	97.9	98.6	99.6	99.7	98.2	98.6	98.5	99.4	99.6	99.8	99.8	93.9	96.2	96.8	97.6	98.8	99.3	99.7
OA _{V0}	99.3	98.9	65.9	78.9	87.2	93.3	95.1	97.1	98.5	90.0	94.0	94.3	96.8	98.6	99.2	99.2	73.8	83.1	88.9	92.6	95.8	96.8	98.4
K ₀	0.99	0.98	0.31	0.62	0.78	0.89	0.92	0.95	0.97	0.83	0.90	0.90	0.95	0.98	0.99	0.99	0.52	0.70	0.81	0.87	0.93	0.95	0.97
OA _{m1}	99.1	99.0	90.3	93.4	96.2	97.7	98.4	98.9	99.0	96.8	98.1	98.2	98.8	98.9	99.1	99.3	92.9	95.3	96.4	97.3	98.4	98.8	98.8
OA _{V1}	96.7	96.4	65.2	78.0	86.7	92.7	94.5	95.7	96.5	86.9	92.6	93.1	94.9	96.5	97.1	97.2	71.1	81.0	87.7	90.8	95.1	95.5	95.4
K ₁	0.94	0.94	0.28	0.60	0.77	0.88	0.90	0.93	0.94	0.76	0.87	0.88	0.91	0.94	0.95	0.95	0.46	0.66	0.79	0.84	0.92	0.92	0.92
OA _{m2}	0.60	0.95	2.51	0.33	0.19	0.34	0.28	0.41	0.34	0.28	0.15	0.15	0.11	0.10	0.29	0.28	1.34	1.17	0.62	0.79	0.14	0.20	0.93
OA _{V2}	2.14	3.45	0.55	0.45	0.30	0.30	0.25	0.60	0.76	5.32	0.68	0.48	1.11	0.77	1.17	1.14	3.30	3.26	2.06	2.77	0.38	0.48	3.40
K ₂	0.060	0.039	0.017	0.008	0.005	0.005	0.005	0.011	0.013	0.143	0.012	0.008	0.019	0.013	0.020	0.020	0.090	0.072	0.042	0.052	0.007	0.009	0.061

RI = reference image; NI = non-atmospheric-corrected image; G(i) = zero-mean Gaussian noise; SP(i) = salt-pepper noise; SPK(i) = speckle noise; OA_{m(r)} = self-model accuracy; OA_{V(r)} = validation accuracy; K_(r) = Cohen's kappa; UA(x) = user's accuracy; PA(x) = production accuracy; x (class) = {1 = agriculture, 2 = bare land, 3 = construction, 4 = water body, 5 = forest}; i = peak signal-to-noise ratio (dB); r = {0 = best, 1 = mean, 2 = standard deviation}.

Table 4. Classification accuracies of the SVM classifiers.

	RI	NI	G10	G15	G20	G25	G30	G35	G40	SP10	SP15	SP20	SP25	SP30	SP35	SP40	SPK10	SPK15	SPK20	SPK25	SPK30	SPK35	SPK40
PA1	99.2	97.6	0	26.2	68.3	88.9	90.5	94.4	97.6	77.0	90.5	95.2	96.8	97.6	98.4	96.0	25.4	49.2	67.5	82.5	91.3	91.3	94.4
PA2	100	98.2	37.3	84.5	87.3	92.7	97.3	100	100	90.9	97.3	98.2	99.1	96.4	99.1	98.2	52.7	73.6	89.1	90.9	98.2	98.2	99.1
PA3	100	95	0	0	0	10.0	40.0	55.0	90.0	10.0	65.0	85.0	85.0	90.0	95.0	95.0	0	0	0	0	40.0	80.0	85.0
PA4	100	100	27.9	76.7	94.2	98.8	100	100	100	87.2	97.7	100	100	100	100	100	89.5	98.8	100	100	100	100	100
PA5	99.6	99.6	94.9	94.3	93.4	96.7	98.0	99.2	99.6	97.5	97.9	99.0	99.4	99.6	99.6	99.4	92.0	94.3	94.9	96.3	97.7	99.0	99.6
UA1	100	98.4	0	62.3	68.3	83.6	91.9	98.3	99.2	85.8	94.2	96.0	98.4	98.4	99.2	98.4	47.8	59.0	70.8	80.6	89.8	95.0	99.2
UA2	98.2	98.2	55.4	84.5	88.1	90.3	93.9	94.8	97.3	90.9	93.0	97.3	98.2	97.2	98.2	98.2	57.4	79.4	84.5	87.7	93.1	94.7	96.5
UA3	100	100	0	0	0	66.7	88.9	100	100	25.0	81.3	100	94.4	94.7	95.0	86.4	0	0	0	0	88.9	100	100
UA4	100	100	68.6	75.9	92.0	97.7	100	100	100	96.2	97.7	100	100	100	100	100	84.6	95.5	100	100	100	100	100
UA5	99.8	99.2	65.2	80.0	90.0	95.7	96.4	97.7	99.4	91.6	97.1	98.4	98.8	99.0	99.6	99.2	79.2	87.0	91.4	94.1	97.1	98.1	98.5
OA _{m0}	99.5	99.4	64.5	77.2	87.2	93.3	98.7	98.1	98.8	89.4	96.2	98.1	98.3	98.7	99.0	99.2	75.9	85.0	88.1	92.5	96.1	97.9	98.8
OA _{V0}	99.6	99.1	64.5	79.0	86.8	93.2	95.7	97.7	99.2	90.5	95.9	98.1	98.7	98.7	99.3	98.7	74.7	83.3	88.4	91.7	95.7	97.4	98.5
K ₀	0.99	0.98	0.23	0.61	0.77	0.88	0.93	0.96	0.99	0.83	0.93	0.97	0.98	0.98	0.99	0.98	0.54	0.70	0.80	0.86	0.93	0.96	0.97
OA _{m1}	99.5	99.4	64.5	77.2	87.2	93.3	98.7	98.1	98.8	89.4	96.2	98.1	98.3	98.7	99.0	99.2	75.9	85.0	88.1	92.5	96.1	97.9	98.8
OA _{V1}	99.6	99.1	64.5	79.0	86.8	93.2	95.4	97.7	99.2	90.4	95.7	98.0	98.4	98.7	99.3	98.7	74.5	83.2	88.3	91.6	95.5	97.3	98.5
K ₁	0.99	0.98	0.23	0.61	0.77	0.88	0.92	0.96	0.99	0.83	0.93	0.97	0.97	0.98	0.99	0.98	0.53	0.70	0.80	0.86	0.92	0.95	0.97
OA _{m2}	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
OA _{V2}	0.02	0.00	0.00	0.04	0.00	0.00	0.06	0.00	0.00	0.06	0.14	0.05	0.12	0.00	0.04	0.00	0.08	0.04	0.06	0.06	0.04	0.07	0.04
K ₂	0.000	0.000	0.000	0.001	0.000	0.000	0.001	0.000	0.000	0.001	0.002	0.001	0.002	0.000	0.001	0.000	0.002	0.001	0.001	0.001	0.001	0.001	0.001

Table 5. Classification accuracies of the RF classifiers.

	RI	NI	G10	G15	G20	G25	G30	G35	G40	SP10	SP15	SP20	SP25	SP30	SP35	SP40	SPK10	SPK15	SPK20	SPK25	SPK30	SPK35	SPK40
PA1	97.6	95.2	10.3	23.8	66.7	84.9	92.9	96.0	94.4	87.3	93.7	94.4	94.4	94.4	95.2	95.2	27.8	57.9	71.4	83.3	92.9	93.7	95.2
PA2	100	100	45.5	81.8	90.0	96.4	97.3	98.2	98.2	95.5	99.1	98.2	99.1	100	99.1	100	50.9	80.0	92.7	93.6	99.1	98.2	97.3
PA3	95	70.0	0	0	0	10.0	20.0	45.0	65.0	0	35.0	60.0	60.0	60.0	70.0	70.0	0	0	0	0	25.0	55.0	65.0
PA4	100	100	31.4	69.8	90.7	98.8	100	98.8	100	98.8	98.8	100	100	100	100	100	88.4	98.8	100	100	100	100	100
PA5	99.4	99.4	88.9	93.4	93.9	96.9	97.9	99.2	99.4	98.8	99.2	99.0	99.0	99.6	99.6	99.4	93.2	94.3	94.7	96.5	97.7	98.8	99.2
UA1	100	99.2	35.1	46.2	71.8	84.3	90.7	97.6	97.5	90.9	97.5	97.5	96.7	99.2	99.2	99.2	41.7	60.8	72.6	82.0	90.0	95.2	96.0
UA2	96.5	94.8	53.2	78.3	85.3	92.2	92.2	95.6	93.9	92.1	92.4	93.9	94.8	93.2	94.8	95.7	57.1	80.7	84.3	88.0	92.4	93.9	94.7
UA3	100	100	0	0	0	66.7	100	81.8	92.9	0	100	92.3	85.7	100	93.3	93.3	0	0	0	0	100	91.7	100
UA4	100	100	50.9	82.2	94.0	97.7	100	100	100	98.8	100	98.9	100	100	100	100	91.6	97.7	100	100	100	100	100
UA5	99.4	98.5	67.9	79.5	89.4	95.0	96.5	97.5	98.5	94.9	97.1	98.1	98.3	98.5	98.6	98.5	81.0	89.8	92.7	94.5	97.1	97.9	98.3
OA _{m0}	99.7	98.9	67.0	78.0	89.5	93.9	96.3	98.0	98.5	95.3	97.9	98.7	98.8	98.9	99.0	98.9	78.0	87.9	88.9	93.6	96.3	98.0	98.8
OA _{V0}	99.2	98.2	63.8	77.8	86.9	93.2	95.4	97.3	97.8	94.4	96.8	97.4	97.5	98.0	98.2	98.2	75.4	85.4	89.3	92.3	95.7	97.1	97.7
K ₀	0.99	0.97	0.28	0.58	0.77	0.88	0.92	0.95	0.96	0.90	0.95	0.96	0.96	0.97	0.97	0.97	0.55	0.75	0.82	0.87	0.93	0.95	0.96
OA _{m1}	99.1	98.5	63.3	76.6	86.8	92.6	95.4	97.3	97.9	94.5	97.0	97.6	97.9	98.3	98.3	98.3	76.7	85.6	88.3	92.5	95.3	97.1	97.9
OA _{V1}	98.8	97.8	62.8	76.7	86.0	92.7	94.9	96.9	97.1	93.8	96.4	97.1	97.0	97.7	97.7	97.8	74.7	84.8	88.7	91.6	95.2	96.5	97.1
K ₁	0.98	0.96	0.25	0.57	0.75	0.87	0.91	0.95	0.95	0.89	0.94	0.95	0.95	0.96	0.96	0.96	0.54	0.74	0.81	0.86	0.92	0.94	0.95
OA _{m2}	0.27	0.25	1.35	1.05	0.91	0.71	0.60	0.41	0.49	0.58	0.47	0.47	0.44	0.33	0.37	0.35	0.79	0.92	0.50	0.62	0.53	0.51	0.46
OA _{V2}	0.19	0.17	0.81	0.53	0.47	0.31	0.28	0.25	0.30	0.28	0.22	0.20	0.29	0.25	0.27	0.25	0.46	0.42	0.30	0.31	0.27	0.27	0.25
K ₂	0.003	0.003	0.016	0.011	0.009	0.005	0.005	0.004	0.005	0.005	0.004	0.003	0.005	0.004	0.005	0.004	0.009	0.007	0.005	0.005	0.005	0.005	0.004

4. Discussion

4.1. Performance of the Classifiers vs. the Reference and the Non-Atmospheric-Corrected Image

The mean kappa coefficients of the same classifier for classification of the reference image and the image without atmospheric correction are very similar. For example, the BPNN produced $K = 0.94$ for both the classification of reference and non-atmospheric-corrected image, which indicates that there is insignificant difference between the classification result of the images. This proves that the classifications using the three ML classifiers on data from a single date Landsat 8 OLI image may not require *absolute* atmospheric correction. In other words, the Level 1 Landsat products are already used for land cover classification after digital conversion to reflectance. However, the absolute atmospheric correction is still required in some works, i.e., time-series analysis and multi-sensor image data.

4.2. Analysis of Noise and Classifiers

Zero-mean Gaussian noise, which was used to illustrate that the effect of adding random values to the image pixels has a significant impact on the classification performance of all classifiers in the presence of high-intensity noise, causes the mean validation accuracies to be lower than 66% ($K \leq 0.28$) (Figure 4). The three classifiers were unable to classify the construction class unless the noise content was decreased to 25 dB (RF and SVM) and 30 dB (BPNN). Moreover, a noise intensity of 10 dB also caused SVM to fail in the classification of the agriculture class, for which BPNN and RF also produced very low accuracies. These phenomena have their origins in the high variance of noise that was added to the training data. This variance causes additional overlapping of the ranges of values among the categories. Another explanation may be that the addition of noise changed the samples significantly. In other words, increasing the additive or zero-mean Gaussian noise decreases the representativeness of each category by changing the pattern of the samples in some or all of the layers.

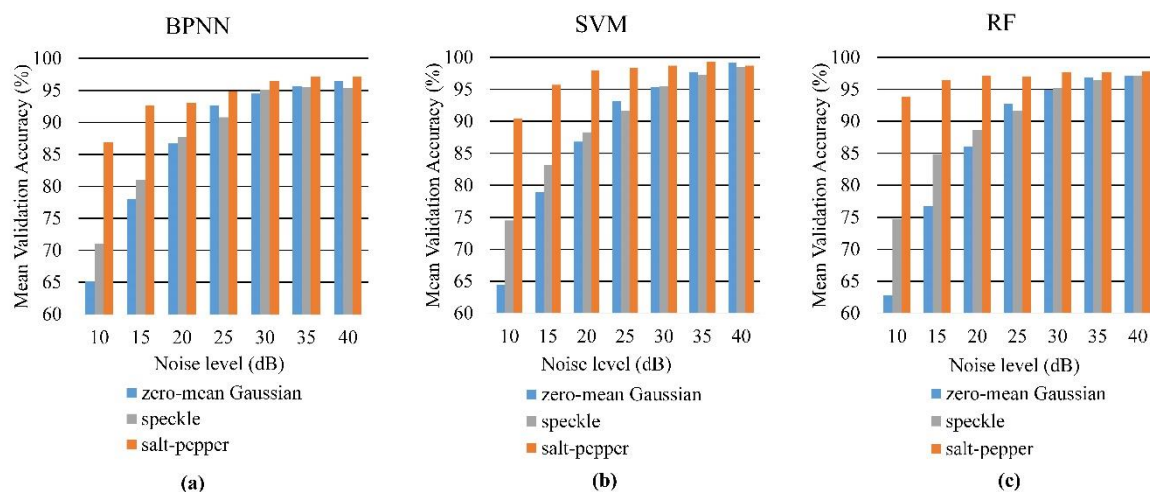


Figure 4. Mean validation accuracy trends of (a) neural network, (b) support vector machine, and (c) random forest classifier in relation to different noise levels and types.

We simulated the occurrence of speckle noise on the reference image by multiplying the random values at different PSNRs. It should be noted that, in real situations, speckle noise either exists infrequently or occurs at very low radiance in Landsat 8 OLI data. The effect of speckle noise is quite similar to that of zero-mean Gaussian noise, in that these types of noise decrease the representativeness of the training model of each class. According to the result, the three algorithms can overcome high-intensity speckle noise more effectively than high-intensity zero-mean Gaussian noise. When considering the UA and PA of each class, the commission and omission errors are lower than the classification accuracies for images to which zero-mean Gaussian noise was added. This is attributed

to the small change in the product after multiplication of the noise and the low pixel value compared to the original when compared to zero-mean Gaussian images, especially in the case of dark objects such as water bodies and mountain shadows (Figure 3).

Salt and pepper noise is the minimum and maximum of the image pixel values (here they are 0 and 1) and randomly replaces the original pixels. In this case, the replaced noise pixels can be seen as data loss or missing data in the case of salt, and black dots in the case of pepper, which can sometimes be removed from the image by using a mask. However, if the reference data points were located among those pixels removed by the mask, it would be problematic, because it would cause valuable ground truth data points to be lost. In this study, the selected ML algorithms were proved to be robust to this kind of noise. Figure 4 clearly shows the mean validation accuracies (OA_{V1}) of all classifiers higher than 85% ($K \geq 0.76$). RF outperforms the other classifiers and has an OA_{V1} value of 93.8% ($K = 0.89$) at the highest noise intensity used in this study (10 dB).

The question may arise here as to why salt–pepper noise had a lesser effect than other types of noise in the classification using these classifiers. This can be explained by pointing to the characteristic of the noise. The noise is not applied to all image pixels; instead, it randomly “replaced” the target pixel by certain values. If the set of a training class is not completely replaced by the noise, the remaining unchanged values can be used to represent the pattern of the class. In other words, we have seven layers for training samples, and if only two of these are changed to the same value (even if they are changed to a different value), the remaining five layers, which are strongly representative of the class, are still available for classification. This is a unique characteristic of salt–pepper noise and is different from that of speckle and zero-mean Gaussian noise.

The algorithm used to train the classifiers also plays an important role in the classification. The black box of basic BPNNs returned a final model that consisted of two main parameters: bias and weights. In this sense, the final model truly represents all of the training samples. This is why the BPNN models trained by noisy data provided lower accuracies than others. Normally, we can improve BPNN accuracy by reducing noise in the data or including more representative training samples in the BPNN learning process. SVM separates binary classes using the optimal hyperplane. The final hyperplane is generated by a few support vectors near the boundary area of the two classes. In this investigation of the effect of noise, the SVM is significantly more robust, because of the fact that the SVM’s parameter selection only determines optimal support vectors to construct the final hyperplanes separating each of the categories. This means that the outlier or noise data were confirmed to be mostly excluded from hyperplane construction. Moreover, for the construction class, which has a very small training size disturbed by high noise intensity, SVM seems to be more effective than the other classifiers. In this study, RF outperformed all the classifiers for the classification of images to which salt–pepper noise was added. This is because the RF randomly generated a large number of trees (decision trees) from training data to produce forests, following which the unknown test samples were voted for by all trees in the final step. This means that anomaly values such as noise may not be voted for by the trees.

In a classification using supervised ML algorithms, the representativeness and the number of images used as training data should be of greater concern when the satellite image is afflicted by noise. Considering the training data in Table 1, the reference samples of each class are imbalanced based on the fact that the percentage of each class in the image is not equal. A rough count of the study area, shown in Figure 2, indicates that the water body and forest class accounts for approximately 60% of the image and the remaining 40% consists of pixels in the bare land, agriculture, and construction classes. An examination of the result of the classifications revealed that the construction class always has the lowest accuracies of any of the classifiers. This may come from one or both of the following: (case 1) the construction class contains too few samples to represent the class characteristic in the presence of noise; (case 2) the limitations of remotely sensed data to represent the class for any given pixel resolution. In this study, based on the results, we can strongly assume that the misclassified samples cannot be explained by case 2, because the accuracy result of the reference image classification confirms that

the construction class can be properly classified. The classification accuracy of the water body and forest classes was mostly high despite the existence of high noise content, because these classes have sufficient training samples to expose the class characteristic even though it is afflicted by noise.

In our experience, supervised-based classifiers, especially ML classifiers, are sensitive to the structure or pattern of training samples because they learn from these data. For example, two size-fixed agricultural areas of a crop with different planting densities (one is sparse, the other is dense) can generally be seen as different classes in medium-resolution images such as a Landsat image due to the effect of the soil background. The user needs to use this information to train the classifier to produce a reasonable and accurate result. This is similar to the case of noise-afflicted data. If it is not possible to prevent or eliminate noise from data, a more effective approach would be to teach the classifier to understand the noise patterns by increasing the number of training samples or by using training data dispersed across the area.

The experiment in this study was carried out on the assumption that the same variance in noise occurred in every layer, whereas the occurrence of real noise varies as it is only found in some bands or as a mixture of different types and intensities of noise. We elucidated the gap by expressing the simulated noise by PSNR as an appropriate way in which the real noise can also be measured by this measurement. However, the intensity of real noise is normally not as high as the noise levels we investigated. The experimental design aimed to determine the effect of noise and to test the performance of the ML algorithms. Consequently, the wide range of noise levels was used to capture the behavior and effects of noise.

4.3. Advanced Extension of the MLs

Recent decades have seen various extensions of the BPNN (e.g., [54–56]), SVM (e.g., [56,57]), and RF (e.g., [56,58]) algorithms that were developed to overcome their drawbacks and, we believe, can perform more effectively than the basic versions of these algorithms. We suggest that readers who require a very high accuracy use the extended version or a noise filter [59–61]. In general, the application of advanced MLs depends greatly on the related theory or technique. For example, fuzzy set concepts [62] are usually attached to basic MLs to make the decision space in categorizing more flexible (fuzzy random forest [63]; fuzzy nonlinear proximal SVM [64]; fuzzy-NN [65]). Markov-systems-based techniques also belong to the leading trend of increasing the classification accuracy (Markov-random-field-based SVM [57]; MLP-Markov chain models [66]). Beside the abovementioned extensions, there are various combinable algorithms that exist in classification field, such as a particle filter [35] and genetic algorithm.

Although the robustness of the basic BPNN towards noises observed in this study was lower than that of RF and SVM, the new-generation neural networks, especially deep learning, are outstanding among other advanced learning algorithms [67]. Deep learning algorithms automatically extract features and abstractions from the underlying data. Numerous reports in the literature have demonstrated that data representations obtained from a deep learning model often yield better results, e.g., improved classification modeling. For example, Hu and Yi (2016) obtained a reasonable result for ground point extraction (for digital terrain modeling) by employing deep convolution neural networks using 17 million labeled Airborne laser scanning (ALS) points [55]. Moreover, Chen et al. (2014) developed a hybrid framework of principle component analysis (PCA), deep learning architecture, and logistic regression for accurate hyperspectral data classification that yielded higher accuracy than RBF-SVM [68]. The examples demonstrated the performance of deep learning over complicated works when a large number of training samples are available.

Nevertheless, the majority of users still use the original algorithms without any complicated extensions. Our work is not only intended to assist private users with the selection of appropriate classifiers for their work, but also to transfer an understanding of remote sensing data to algorithm developers. Future work on other ML classifiers, especially on new algorithms invented in the last decade, would be necessary to compare them to the existing algorithms.

5. Conclusions

In this study, we investigated the effects of adding simulated zero-mean Gaussian, salt-pepper, and speckle noise on image classification performance. Three well-known machine-learning classifiers, i.e., back-propagation neural network (BPNN), support vector machine (SVM), and random forest (RF), with the most basic parameter settings, were used for the classifications. A Landsat 8 OLI scene of Srinakarin dam, Kanchanaburi Province, Thailand, acquired under cloud-free conditions with absolute atmospheric correction, was used as reference data for noise affliction. We specified the same number of usable training data for all classifiers. The experiment started by classifying the reference image by the classifiers, after which the results were set as the baseline accuracy for interpretation. The reference image was then contaminated by different levels of noise at peak signal-to-noise ratios (PSNRs) ranging from 10 to 40 dB (in increments of 5 dB). Finally, 21 noisy images were obtained and separately classified by the classifiers. As expected, the lower the PSNR (high noise intensity), the lower the accuracy. All classifiers provided accuracies over 96% for the reference and the image without atmospheric correction. This means that absolute atmospheric correction in single date classification is unnecessary for images acquired under clear weather conditions. We suggest from our experiment that an increase in the number of training and collecting samples dispersed in various patterns across the study area facilitates the improvement of the representativeness of the class when the data is afflicted by high-intensity noise. More specifically, in terms of the performance of the classifiers, the BPNN models provided accuracies lower than others in very high noise intensity, because the algorithm strongly depends on the values of the training samples. According to the results, SVM often provides a high accuracy because SVM parameter selection can be used to ensure that only optimal support vectors are found to construct the final hyperplanes separating each of the categories, and the minor fluctuations caused by noise were confirmed to mostly be excluded from model construction. RF outperformed all of the classifiers in salt-pepper noise added image classifications. However, although in this study the three ML classifiers proved to be very robust to low-to-medium noise intensity (PSNR > 20) without using any filter or extension, we suggest taking advantage of a noise filter or the advanced extensions of these ML classifiers, especially for cases requiring very high accuracy (almost 100%).

Author Contributions: Conceptualization, S.B.; methodology, S.B.; software, S.B. and B.K.A.; validation, S.B. and W.C.; formal analysis, S.B.; investigation, C.C.; resources, C.C. and M.X.; data Curation, S.B. and W.C.; writing—original draft preparation, S.B.; writing—review and editing, S.B.; visualization, S.B. and B.K.A.; supervision, C.C. and W.C.; project administration, C.C., M.X., and X.N.; funding acquisition, C.C.

Funding: This research was financed by the Natural Science Foundation of China (No. 41601368), the National Key Research and Development Program of China (No. 2016YFB0501505), and the Instrument Development Project of the State Key Laboratory of Remote Sensing Science (No. Y7Y01100KZ).

Acknowledgments: The authors would like to extend their gratitude to the Geo-Informatics and Space Technology Development Agency (GISTDA), Thailand, for the reference data, and to Peerapong Torteeka for his assistance in revising the algorithms. Sornkitja Boonprong would like to express his gratitude to the Royal Thai Government for their support and the scholarship provided. The authors would also like to thank the anonymous reviewers for their insightful comments, which have significantly improved this paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Song, M.; Civco, D.L.; Hurd, J.D. A competitive pixel-object approach for land cover classification. *Int. J. Remote Sens.* **2005**, *26*, 4981–4997. [[CrossRef](#)]
2. Petrou, Z.I.; Kosmidou, V.; Manakos, I.; Stathaki, T.; Adamo, M.; Tarantino, C.; Tomaselli, V.; Blonda, P.; Petrou, M. A rule-based classification methodology to handle uncertainty in habitat mapping employing evidential reasoning and fuzzy logic. *Pattern Recognit. Lett.* **2014**, *48*, 24–33. [[CrossRef](#)]
3. Simonetti, E.; Simonetti, D.; Preatoni, D. *Phenology-Based Land Cover Classification Using Landsat 8 Time Series*; European Commission Joint Research Center: Ispra, Italy, 2014.

4. Loosvelt, L.; Peters, J.; Skriver, H.; Lievens, H.; Van Coillie, F.M.B.; De Baets, B.; Verhoest, N.E.C. Random Forests as a tool for estimating uncertainty at pixel-level in SAR image classification. *Int. J. Appl. Earth Obs.* **2012**, *19*, 173–184. [CrossRef]
5. Al-amri, S.; Kalyankar, N.; Khamitkar, S. A Comparative Study of Removal Noise from Remote Sensing Image. *IJCSI Int. J. Comput. Sci.* **2010**, *7*, 32–36.
6. Dierking, W.; Dall, J. Sea Ice Deformation State from Synthetic Aperture Radar Imagery—Part II: Effects of Spatial Resolution and Noise Level. *IEEE Trans. Geosci. Remote Sens.* **2008**, *46*, 2197–2207. [CrossRef]
7. Barber, M.; Grings, F.; Perna, P.; Piscitelli, M.; Maas, M.; Bruscantini, C.; Jacobo-Berlles, J.; Karszenbaum, H. Speckle Noise and Soil Heterogeneities as Error Sources in a Bayesian Soil Moisture Retrieval Scheme for SAR Data. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2012**, *5*, 942–951. [CrossRef]
8. Benz, U.; Hofmann, P.; Willhauck, G.; Lingenfelder, I.; Heynen, M. Multi-resolution, object-oriented fuzzy analysis of remote sensing data for GIS-ready information. *ISPRS J. Photogramm. Remote Sens.* **2004**, *58*, 239–258. [CrossRef]
9. Celik, T.; Ma, K.K. Unsupervised Change Detection for Satellite Images Using Dual-Tree Complex Wavelet Transform. *IEEE Trans. Geosci. Remote Sens.* **2010**, *48*, 1199–1210. [CrossRef]
10. Mohammed, K.; Mohamed, D.; Philippe, C. Magnitude-phase of the dual-tree quaternionic wavelet transform for multispectral satellite image denoising. *EURASIP J. Image Video Process.* **2014**, *2014*, 1–16.
11. Vijay, M.; Devi, L.S. Speckle Noise Reduction in Satellite Images Using Spatially Adaptive Wavelet Thresholding. *Int. J. Comput. Sci. Inf. Technol.* **2012**, *3*, 3432–3435.
12. Bhosale, N.P.; Manza, R.R. Analysis of Effect of Noise Removal Filters on Noisy Remote Sensing Images. *Int. J. Sci. Eng. Res.* **2013**, *4*, 1151.
13. Data Loss, Landsat Missions, USGS. Available online: <https://landsat.usgs.gov/data-loss> (accessed on 5 June 2018).
14. Charoenjit, K.; Zuddas, P.; Allemand, P.; Pattanakit, S.; Pachana, K. Estimation of biomass and carbon stock in Para rubber plantations using object-based classification from Thaichote satellite data in Eastern Thailand. *J. Appl. Remote Sens.* **2015**, *9*, 096072. [CrossRef]
15. Yahya, N.; Kamel, N.S.; Malik, A.S. Subspace-Based Technique for Speckle Noise Reduction in SAR Images. *IEEE Trans. Geosci. Remote Sens.* **2014**, *52*, 6257–6271. [CrossRef]
16. Raney, R.K. Radar Fundamentals: Technical Perspective. In *Principles and Applications of Imaging Radar, Manual of Remote Sensing*, 3rd ed.; John Wiley and Sons Inc.: Toronto, ON, Canada, 1998; Chapter 2.
17. Mansourpour, M.; Rajabi, M.A.; Blais, J.A.R. Effects and Performance of Speckle Noise Reduction Filters on Active Radar and SAR Images. *Proc. ISPRS* **2006**, *36*, W41.
18. Nobrega, R.A.A.; Quintanilha, J.A.; O'Hara, C.G. A noise-removal approach for LIDAR intensity images using anisotropic diffusion filtering to preserve object shape characteristics. In Proceedings of the ASPRS 2007 Annual Conference, Tampa, FL, USA, 7–11 May 2007.
19. Mulder, V.L.; de Bruin, S.; Schaepman, M.E.; Mayr, T.R. The use of remote sensing in soil and terrain mapping—A review. *Geoderma* **2011**, *162*, 1–19. [CrossRef]
20. Zhu, X.; Wu, X. Class Noise vs. Attribute Noise: A Quantitative Study of Their Impacts. *Artif. Intell. Rev.* **2004**, *22*, 177–210. [CrossRef]
21. Breiman, L. Random Forests. *Mach. Learn.* **2001**, *45*, 5–32. [CrossRef]
22. Alsmadi, M.K.S.; Omar, K.B.; Noah, S.A. Back Propagation Algorithm: The Best Algorithm among the Multi-layer Perceptron Algorithm. *Int. J. Comput. Sci. Netw. Secur.* **2009**, *9*, 378–383.
23. Caruana, R.; Niculescu-Mizil, A. An Empirical Comparison of Supervised Learning Algorithms. In Proceedings of the 23rd International Conference on Machine Learning, Pittsburgh, PA, USA, 25–29 June 2006; pp. 161–168.
24. Dey, A. Machine Learning Algorithms: A Review. *Int. J. Comput. Sci. Inf. Technol.* **2016**, *7*, 1174–1179.
25. Kotsiantis, S.B. Supervised Machine Learning: A Review of Classification Techniques. *Informatica* **2007**, *31*, 249–268.
26. Singh, A.; Thakur, N.; Sharma, A. A review of supervised machine learning algorithms. In Proceedings of the 2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, India, 16–18 March 2016; pp. 1310–1315.
27. Crisci, C.; Ghattas, B.; Perera, G. A review of supervised machine learning algorithms and their applications to ecological data. *Ecol. Model.* **2012**, *240*, 113–122. [CrossRef]

28. Lorena, A.C.; Jacintho, L.F.O.; Siqueira, M.F.; Giovanni, R.D.; Lohmann, L.G.; de Carvalho, A.C.P.L.F.; Yamamoto, M. Comparing machine learning classifiers in potential distribution modelling. *Expert Syst. Appl.* **2011**, *38*, 5268–5275. [[CrossRef](#)]
29. Mariana, B.; Lucian, D. Random forest in remote sensing: A review of application and future directions. *ISPRS J. Photogramm. Remote Sens.* **2016**, *114*, 24–31.
30. Rodriguez-Galiano, V.F.; Ghimire, B.; Rogan, J.; Chica-Olmo, M.; Rigol-Sanchez, J.P. An assessment of the effectiveness of a random forest classifier for land-cover classification. *ISPRS J. Photogramm. Remote Sens.* **2012**, *67*, 93–104. [[CrossRef](#)]
31. Loosvelt, L.; Peters, J.; Skriver, H.; De Baets, B.; Verhoest, N.E.C. Impact of Reducing Polarimetric SAR Input on the Uncertainty of Crop Classifications Based on the Random Forests Algorithm. *IEEE Trans. Geosci. Remote Sens.* **2012**, *50*, 4185–4200. [[CrossRef](#)]
32. Freund, Y.; Schapire, R.E. Experiments with a new boosting algorithm. In Proceedings of the Thirteenth International Conference on International Conference on Machine Learning, Bari, Italy, 3–6 July 1996; pp. 148–156.
33. Mountrakis, G.; Im, J.; Ogole, C. Support vector machines in remote sensing: A review. *ISPRS J. Photogramm. Remote Sens.* **2011**, *66*, 247–259. [[CrossRef](#)]
34. Insom, P.; Cao, C.; Boonsrimuang, P.; Liu, D.; Saokarn, A.; Yomwan, P.; Xu, Y. A Support Vector Machine-Based Particle Filter Method for Improved Flooding Classification. *IEEE Geosci. Remote Sens. Lett.* **2015**, *12*, 1943–1947. [[CrossRef](#)]
35. Foody, G.M.; Mathur, A. Toward intelligent training of supervised image classifications: Directing training data acquisition for SVM classification. *Remote Sens. Environ.* **2004**, *93*, 107–117. [[CrossRef](#)]
36. Dalponte, M.; Bruzzone, L.; Gianelle, D. Fusion of Hyperspectral and LIDAR Remote Sensing Data for Classification of Complex Forest Areas. *IEEE Trans. Geosci. Remote Sens.* **2008**, *46*, 1416–1427. [[CrossRef](#)]
37. Senf, C.; Leitao, P.J.; Pflugmacher, D.; Linden, S.V.D.; Hostert, P. Mapping land cover in complex Mediterranean landscapes using Landsat: Improved classification accuracies from integrating multi-seasonal and synthetic imagery. *Remote Sens. Environ.* **2015**, *156*, 527–536. [[CrossRef](#)]
38. Pal, M.; Foody, G.M. Feature Selection for Classification of Hyperspectral Data by SVM. *IEEE Trans. Geosci. Remote Sens.* **2010**, *48*, 2297–2307. [[CrossRef](#)]
39. Cherkassky, V.; Ma, Y. Practical selection of SVM parameters and noise estimation for SVM regression. *Neural Netw.* **2004**, *17*, 113–126. [[CrossRef](#)]
40. Tseng, M.H.; Chen, S.J.; Hwang, G.H.; Shen, M.Y. A genetic algorithm rule-based approach for land-cover classification. *ISPRS J. Photogramm. Remote Sens.* **2008**, *63*, 202–212. [[CrossRef](#)]
41. Suliman, A.; Zhang, Y. A Review on Back-Propagation Neural Networks in the Application of Remote Sensing Image Classification. *J. Earth Sci. Eng.* **2015**, *5*, 52–65.
42. Paola, J.D.; Schowengerdt, R.A. A review and analysis of backpropagation neural networks for classification of remotely-sensed multi-spectral imagery. *Int. J. Remote Sens.* **2007**, *16*, 3033–3058. [[CrossRef](#)]
43. Burzzone, L.; Fernandez, P.D.; Cossu, R. An incremental learning classifier for remote-sensing images. In Proceedings of the IEEE 1999 International Geoscience and Remote Sensing Symposium, Hamburg, Germany, 28 June–2 July 1999; pp. 2483–2485.
44. Avanaki, M.R.N.; Laissue, P.; Podoleanu, A.; Hojjat, S.A. Denoising based on noise parameter estimation in speckled OCT images using neural network. In Proceedings of the 1st Canterbury Workshop and School in Optical Coherence Tomography and Adaptive Optics, Canterbury, UK, 30 December 2008. [[CrossRef](#)]
45. Kofidis, E.; Theodoridis, S.; Kotropoulos, C.; Pitas, I. Application of Neural Networks and Order Statistics Filters to Speckle Noise Reduction in Remote Sensing Imaging. In *Neurocomputation in Remote Sensing Data Analysis*; Springer: Berlin/Heidelberg, Germany, 1997. [[CrossRef](#)]
46. Silva, I.B.V.D.; Adeodato, P.J.L. PCA and Gaussian noise in MLP neural network training improve generalization in problems with small and unbalanced data sets. In Proceedings of the 2011 International Joint Conference on Neural Networks, San Jose, CA, USA, 31 July–5 August 2011; pp. 2664–2669. [[CrossRef](#)]
47. Corner, B.R.; Narayanan, R.M.; Reichenbach, S.E. Noise estimation in remote sensing imagery using data masking. *Int. J. Remote Sens.* **2003**, *24*, 689–702. [[CrossRef](#)]
48. Breiman, L. *Manual on Setting Up, Using, and Understanding Random Forests v3.1*; Department of Statistics, University of California: Berkeley, CA, USA, 2002.

49. Moller, M.F. A scaled conjugate gradient algorithm for fast supervised learning. *Neural Netw.* **1993**, *6*, 525–533. [[CrossRef](#)]
50. Boser, B.E.; Guyon, I.M.; Vapnik, V.N. A training algorithm for optimal margin classifiers. In Proceedings of the Fifth Annual Workshop on Computational Learning Theory, Pittsburgh, PA, USA, 27–29 July 1992; pp. 144–152.
51. Cortes, C.; Vapnik, V. Support-vector networks. *Mach. Learn.* **1995**, *20*, 273–297. [[CrossRef](#)]
52. Liaw, A.; Wiener, M. Classification and Regression by Random Forest. *R News* **2002**, *2*, 18–22.
53. Chang, C.C.; Lin, C.J. LIBSVM: A library for support vector machine. *ACM Trans. Intell. Syst. Technol.* **2011**, *2*, 27. [[CrossRef](#)]
54. Rodríguez-Fernández, N.J.; Kerr, Y.H.; van der Schalie, R.; Al-Yaari, A.; Wigneron, J.P.; de Jeu, R.; Richaume, P.; Dutra, E.; Mialon, A.; Drusch, M. Long Term Global Surface Soil Moisture Fields Using an SMOS-Trained Neural Network Applied to AMSR-E Data. *Remote Sens.* **2016**, *8*, 959. [[CrossRef](#)]
55. Hu, X.; Yuan, Y. Deep-Learning-Based Classification for DTM Extraction from ALS Point Cloud. *Remote Sens.* **2016**, *8*, 730. [[CrossRef](#)]
56. Li, X.; Chen, W.; Cheng, X.; Wang, L. A Comparison of Machine Learning Algorithms for Mapping of Complex Surface-Mined and Agricultural Landscapes Using ZiYuan-3 Stereo Satellite Imagery. *Remote Sens.* **2016**, *8*, 514. [[CrossRef](#)]
57. Yu, H.; Gao, L.; Li, J.; Li, S.S.; Zhang, B.; Benediktsson, J. Spectral-Spatial Hyperspectral Image Classification Using Subspace-Based Support Vector Machines and Adaptive Markov Random Fields. *Remote Sens.* **2016**, *8*, 355. [[CrossRef](#)]
58. Wessels, K.J.; van den Bergh, F.; Roy, D.P.; Salmon, B.P.; Steenkamp, K.C.; MacAlister, B.; Swanepoel, D.; Jewitt, D. Rapid Land Cover Map Updates Using Change Detection and Robust Random Forest Classifiers. *Remote Sens.* **2016**, *8*, 888. [[CrossRef](#)]
59. Shrestha, R.; Wynne, R.H. Estimating biophysical parameters of individual trees in an urban environment using small footprint discrete-return imaging Lidar. *Remote Sens.* **2012**, *4*, 484–508. [[CrossRef](#)]
60. Boonprong, S.; Cao, C.; Torteeka, P.; Chen, W. A Novel Classification Technique of Landsat-8 OLI Image-Based Data Visualization: The Application of Andrews' Plots and Fuzzy Evidential Reasoning. *Remote Sens.* **2017**, *9*, 427. [[CrossRef](#)]
61. Chang, E.S.; Hung, C.C.; Liu, W.; Yin, J. A Denoising Algorithm for Remote Sensing Images with Impulse Noise. In Proceedings of the IEEE International Geoscience and Remote Sensing Symposium (IGARSS), Beijing, China, 10–15 July 2016; pp. 2905–2908.
62. Zadeh, L.A. Fuzzy Sets. *Inf. Control* **1965**, *8*, 338–355. [[CrossRef](#)]
63. Bonissone, P.; Cadenas, J.M.; Garrido, M.C.; Díaz-Valladares, R.A. A fuzzy random forest. *Int. J. Approx. Reason.* **2010**, *51*, 729–747. [[CrossRef](#)]
64. Zhong, X.; Li, J.; Dou, H.; Deng, S.; Wang, G.; Jiang, Y.; Wang, Y.; Zhou, Z.; Wang, L.; Yan, F. Fuzzy Nonlinear Proximal Support Vector Machine for Land Extraction Based on Remote Sensing Image. *PLoS ONE* **2013**, *8*, e69434. [[CrossRef](#)] [[PubMed](#)]
65. Goodarzi, R.; Mokhtarzade, M.; Zoj, M.J.V. A Robust Fuzzy Neural Network Model for Soil Lead Estimation from Spectral Features. *Remote Sens.* **2015**, *7*, 8416–8435. [[CrossRef](#)]
66. Ozturk, D. Urban Growth Simulation of Atakum (Samsun, Turkey) Using Cellular Automata-Markov Chain and Multi-Layer Perceptron-Markov Chain Models. *Remote Sens.* **2015**, *7*, 5918–5950. [[CrossRef](#)]
67. Najafabadi, M.M.; Villanustre, F.; Khoshgoftaar, T.M.; Seliya, N.; Wald, R.; Muharemagic, E. Deep learning applications and challenges in big data analytics. *J. Big Data* **2015**, *2*, 1–21. [[CrossRef](#)]
68. Chen, Y.; Lin, Z.; Zhao, X.; Wang, G.; Gu, Y. Deep Learning-Based Classification of Hyperspectral Data. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2014**, *7*, 2094–2107. [[CrossRef](#)]

