ISPRS Open Journal of Photogrammetry and Remote Sensing 4 (2022) 100012

Contents lists available at ScienceDirect

ISPRS Open Journal of Photogrammetry and Remote Sensing

journal homepage: www.journals.elsevier.com/isprs-open-journal-of-photogrammetry-and-remote-sensing

# Comparison of neural networks and k-nearest neighbors methods in forest stand variable estimation using airborne laser data

Andras Balazs, Eero Liski, Sakari Tuominen *, Annika Kangas

*Natural Resources Institute, Finland*

ARTICLE INFO

*Keywords:*
Deep learning
Artificial neural network
Convolutional neural network
Machine learning
Remote sensing
Forest inventory
Airborne laser scanning

ABSTRACT

In the remote sensing of forests, point cloud data from airborne laser scanning contains high-value information for predicting the volume of growing stock and the size of trees. At the same time, laser scanning data allows a very high number of potential features that can be extracted from the point cloud data for predicting the forest variables. In some methods, the features are first extracted by user-defined algorithms and the best features are selected based on supervised learning, whereas both tasks can be carried out automatically by deep learning methods typically based on deep neural networks. In this study we tested k-nearest neighbor method combined with genetic algorithm (k-NN), artificial neural network (ANN), 2-dimensional convolutional neural network (2D-CNN) and 3-dimensional CNN (3D-CNN) for estimating the following forest variables: volume of growing stock, stand mean height and mean diameter. The results indicate that there were no major differences in the accuracy of the tested methods, but the ANN and 3D-CNN generally resulted in the lowest RMSE values for the predicted forest variables and the highest $R^2$ values between the predicted and observed forest variables. The lowest RMSE scores were 20.3% (3D-CNN), 6.4% (3D-CNN) and 11.2% (ANN) and the highest $R^2$ results 0.90 (3D-CNN), 0.95 (3D-CNN) and 0.85 (ANN) for volume of growing stock, stand mean height and mean diameter, respectively. Covariances of all response variable combinations and all predictions methods were lower than corresponding covariances of the field observations. ANN predictions had the highest covariances for mean height vs. mean diameter and total growing stock vs. mean diameter combinations and 3D-CNN for mean height vs. total growing stock. CNNs have distinct theoretical advantage over the other methods in complex recognition or classification tasks, but the utilization of their full potential may possibly require higher point density clouds than applied here. Thus, the relatively low density of the point clouds data may have been a contributing factor to the somewhat inconclusive ranking of the methods in this study. The input data and computer codes are available at: https://github.com/balazsan/ALS_NNs.

## 1. Introduction

Airborne laser scanning (ALS) data have been utilized from the early 2000s in forest stand variable estimation (Lefsky et al., 2002; Næsset, 2002; Maltamo et al., 2004). Traditionally the interpretation of laser data was carried out using ALS features such as height metrics derived from above ground heights of laser returns (Næsset, 2004; Packalén and Maltamo, 2006; Packalén et al., 2009). ALS features serve as explanatory variables in modeling forest stand variables, e.g., total growing stock or mean height (Packalén and Maltamo, 2007; Tuominen et al., 2014) .

Due to the popularity of deep learning techniques in recent years, once again Neural Networks (NNs) have become the subject of attention and were found to be useful in forestry related applications (Ayrey and Hayes, 2018; Hamraz et al., 2018; Fricker et al., 2019). NNs are composed of nodes (neurons) and connections between the neurons (Shukla and Fricklas, 2018). Neurons are usually organized in layers. A simple NN can consist of an input and output layer with one or several hidden layers in-between.

Artificial Neural Network (ANN) is the simplest form of NNs. Typically, explanatory variables used with ANNs are derived from remote sensing datasets, e.g., ALS or aerial image features. ANNs have been used in automatic detection of wildfires (Li et al., 2001), Leaf Area Index estimation (Jensen and Binford, 2004), predicting stand level forest attributes (Pohjankukka et al., 2018), as well as predicting attributes of individual trees (Tavares Júnior et al., 2019; Corte et al., 2020).

Unlike ANNs, Convolutional Neural Networks (CNNs), a member of

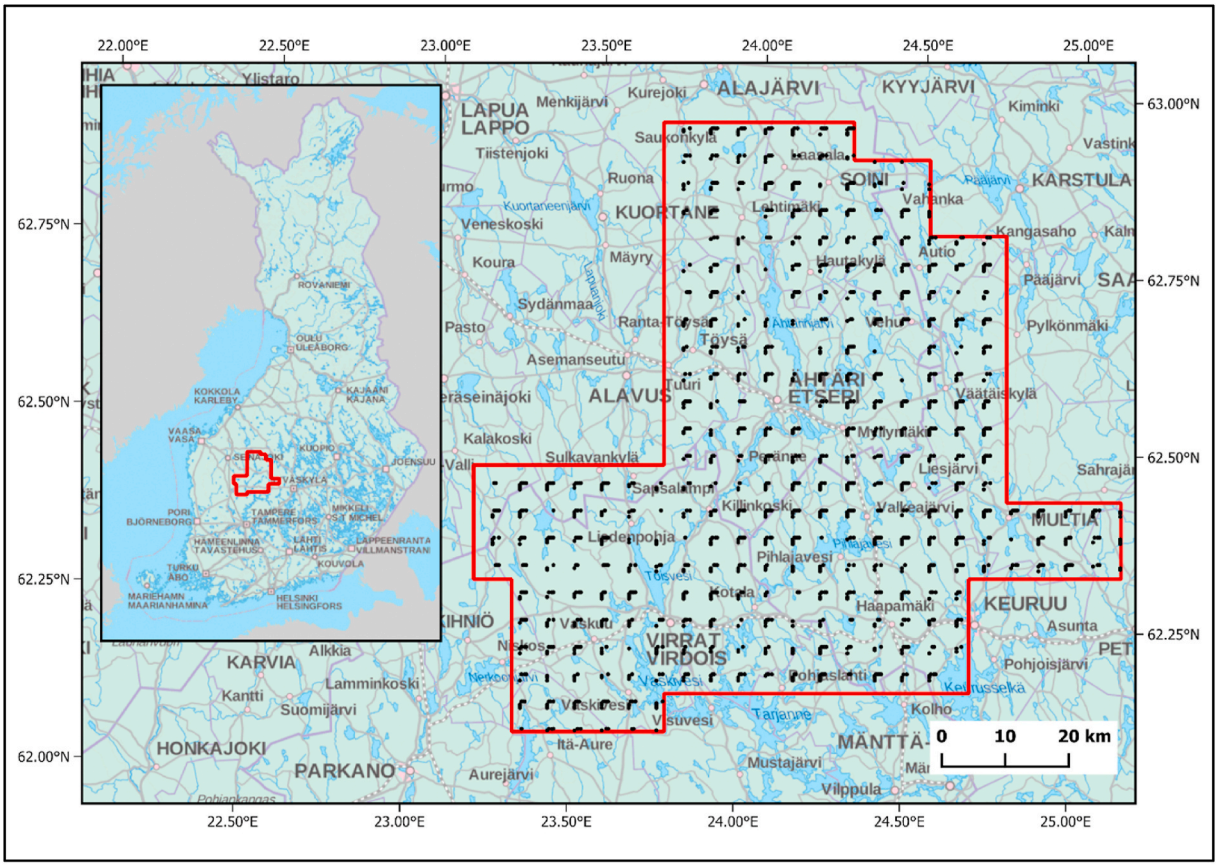
* Corresponding author.
*E-mail addresses:* andras.balazs@luke.fi (A. Balazs), eero.liski@luke.fi (E. Liski), sakari.tuominen@luke.fi (S. Tuominen), annika.kangas@luke.fi (A. Kangas).

https://doi.org/10.1016/j.ophoto.2022.100012
Received 16 September 2021; Received in revised form 16 February 2022; Accepted 18 February 2022
Available online 8 March 2022

the deep learning family, are complicated computational systems generally known from computer vision tasks like face recognition. In the remote sensing of forests, CNNs have been utilized for classification tasks such as vegetation type and tree species recognition (Syrris et al., 2019; Illarionova et al., 2021; Moura et al., 2021) as well as prediction of quantitative variables (Ayrey and Hayes, 2018; Shah et al., 2020; Liu et al., 2021). CNNs do not require user-specified input features, which results in lighter preprocessing of the input data. CNNs can recognize complex relationships between explanatory and response variables that could be hidden from feature architects. In CNNs convolutional windows scan the input data and extract relevant features on various levels (Shukla and Fricklas, 2018).

CNNs can have 1–3 dimensions. One dimension could represent spectral variation, two dimensions spatial variation in an image and three dimensions either a combination of the previous two or a point cloud. Ayrey and Hayes (2018) used three-dimensional CNNs (3D CNNs) successfully in estimating forest attributes. Two-dimensional CNNs (2D CNN) have been used e.g., for tree species recognition by Fricker et al. (2019) and coniferous/deciduous tree classification by Hamraz et al. (2018).

Due to the high number of tunable parameters in deep learning models, their training usually requires a large amount of data (Krizhevsky et al., 2017). For example, Ayrey and Hayes (2018) used over 17 000 ground reference plots collected in eight different areas. Such an excessive reference dataset, however, is rarely available due to the financial aspect of data collection. Data augmentation is a widely used




*Map of the study area. The laser scanning area is outlined with red line; locations of sample plots are marked with black dots. The background map is the raster map 1:800 000 by the National Land Survey of Finland.*

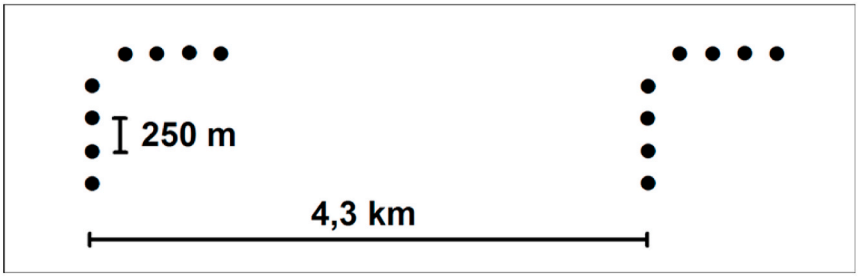



*Illustration of the sample plot layout within the clusters.*

**Fig. 1.** **a** Map of the study area. The laser scanning area is outlined with red line; locations of sample plots are marked with black dots. The background map is the raster map 1:800 000 by the National Land Survey of Finland.
**b** Illustration of the sample plot layout within the clusters. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

solution to inflate the size and increase the diversity of limited training data (Shorten and Khoshgoftaar, 2019). For computer vision tasks rotating images is a common augmentation technique which can also be applied to laser data.

The objective of this study was to determine whether three different types of NNs can outperform our benchmark k-nearest neighbors (k-NN) method in estimating forest stand attributes. The following NNs were included from simple to more complex in structure: fully connected ANN with one hidden layer, 2-dimensional CNN (2D-CNN) based on AlexNet and 3-dimensional CNN (3D-CNN) based on Inception V3 by Ayrey and Hayes (2018). Given the limited amount of reference data we also wanted to discover whether it is possible to successfully train CNNs with a relatively small set of training samples. Furthermore, we wanted to find out if the correlation structure between different predicted variables can be retained.

## 2. Materials

### 2.1. Study area and reference data

The study area is located in central Finland and covers approximately 5800 km$^2$. The topography of the area is relatively flat with elevation values ranging generally between 100 and 200 m. The study area is characterized by forested areas, sparse inhabitation, and lack of urban areas. The forests in the area mainly grow on oligotrophic moraine and peat soil deposits, which have been unfavorable for settlement and cultivation.

The map of the study area sample plot layout are pictured in Fig. 1a and b respectively.

The reference data used was part of a sample plot network created for a study by Tomppo et al. (2016). Sample plots were systematically distributed over the study area using clusters of 8 sample plots with 4.3 km and 250 m spacing between clusters and sample plots respectively. Trees with at least 4.5 cm diameter at breast height (DBH) were measured within 9 m radius in each plot. Tree species, DBH, tree class (sawtimber, pulp, non-merchantable or dead tree) and position of all trees were recorded. Additionally, heights were measured for basal-area median trees in tree species groups in each sample plot to enable tree height imputation for all tallied trees.

1494 sample plots were selected from the original dataset for this study. The criteria for inclusion were for the plots to be situated in forestry land and during leaf-on season scanned airborne laser data covering the plot. Both laser scanning and field measurements were carried out in 2013.

Statistics of the forest stand variables relevant to this study are shown in Table 1.

It is a common approach to divide data for training neural networks into training and validation sets (e.g. Ayrey and Hayes, 2018). The training set is used to train the network and the validation set to monitor the network's performance during training at an interval of a predefined amount of training steps. Only the best performing model is saved to disk. To have an independent set of data to evaluate performances of all methods a test set was created additionally to training and validation sets. These subsets were created in a way that the distribution of total growing stock volume was approximately the same in each set. The training set comprised of 1044, each of the validation and test sets of 225 sample plots. Fig. 2 shows the distribution of the reference sub-datasets in classes of total growing stock volume.

Forest variables of all sub-datasets (training, validation and test) were scaled to the mean and standard deviation of the training dataset. Scaling may be useful (Bishop, 1996) as different scales across the variables to be estimated increase the complexity of the problem.

**Table 1**
Plot summary statistics.

| | Total growing stock (m$^3$/ha) | Mean height (m) | Mean diameter (cm) |
|---|---|---|---|
| Mean | 147.4 | 14.7 | 18.0 |
| Standard deviation | 89.3 | 4.2 | 5.4 |
| Max | 867.4 | 28.1 | 43.8 |

### 2.2. Airborne laser scanning data

ALS data was acquired during the summer of 2013 using Optech's ALTM Gemini laser scanner. The scanning was carried out from an altitude of 1730 m with a maximum zenith angle of 20° and a side overlap of 20% and a maximum of 4 pulse returns were captured. The resulting point cloud had an average point density of 0.91 m$^2$.

The point cloud was first clipped applying a 9 m radius circular buffer around the locations of reference field plots' centers. Point heights were then normalized to above ground heights using a digital terrain model at 2 m resolution provided by the National Land Survey of Finland.

CNNs require that the input data has standard dimensions for each sample. Due to this requirement circular point clouds were transformed into voxel spaces. The number of voxels in each voxel space was predefined for the 3D-CNN by Ayrey and Hayes (2018). Each voxel space measured 18 × 18 × 41 m with 40 × 40 × 105 voxels resulting in a voxel size of 45 × 45 × 39 cm in x, y, z directions respectively. Similarly, to the ground truth dataset, voxel data was divided into training, validation and test sets. Three types of voxel values were tested initially; number of points (ALS returns) in voxel, normalized and binary voxels. Min-max normalization of number-of-points voxel values in all subsets was carried out using the training set's minimum and maximum. In case of binary voxels each voxel takes value 0 if it does not contain any laser returns or 1 if at least one laser return is within the voxel. Fig. 3 shows an illustration of a voxel space with increased voxel size for visibility with legend according to height above ground.

To keep the augmentation process simple, the training dataset's voxel spaces were randomly rotated by 90, 180 or 270° around the z-axis resulting in a training data size of 2088 sample plots. Randomness was applied not to artificially create a pattern in the training data. Pre-augmented training data was used for the training of the 2D-CNN, whereas augmentation was incorporated in the code of the 3D-CNN.

## 3. Methods

A summary of the workflow is presented in Fig. 4. The individual steps are discussed in detail in the following sections.

### 3.1. Point cloud metrics

Remote sensing features were extracted from the ALS point clouds. The features can be divided into two groups, height metrics based on the above ground heights of laser returns and texture features based on rasterized above ground heights of first laser returns. The features were extracted from 16 × 16 m window around the sample plot centers at 0.5 m resolution.

Additionally to standard height metrics calculated using lidR R-package (Roussel, 2021) the following features were extracted (Næsset, 2002; St-Onge et al., 2008; Véga et al., 2016; Kirchhoefer et al., 2017):

- percentage of canopy points having height greater than or equal to corresponding percentile of height distribution, at percentiles 20, 40, 60, 80, 95,
- canopy relief ratio,
- L-moments (1–4) and L-moment coefficient of variation, skewness and kurtosis,
- quadratic and cubic mean,
- average absolute deviation,

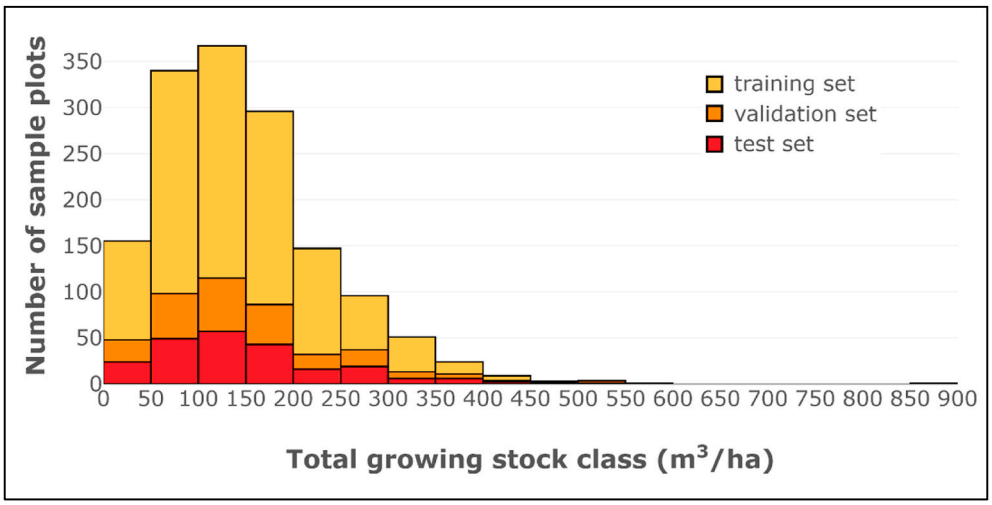



**Fig. 2.** Distribution of sample plots in total growing stock classes.

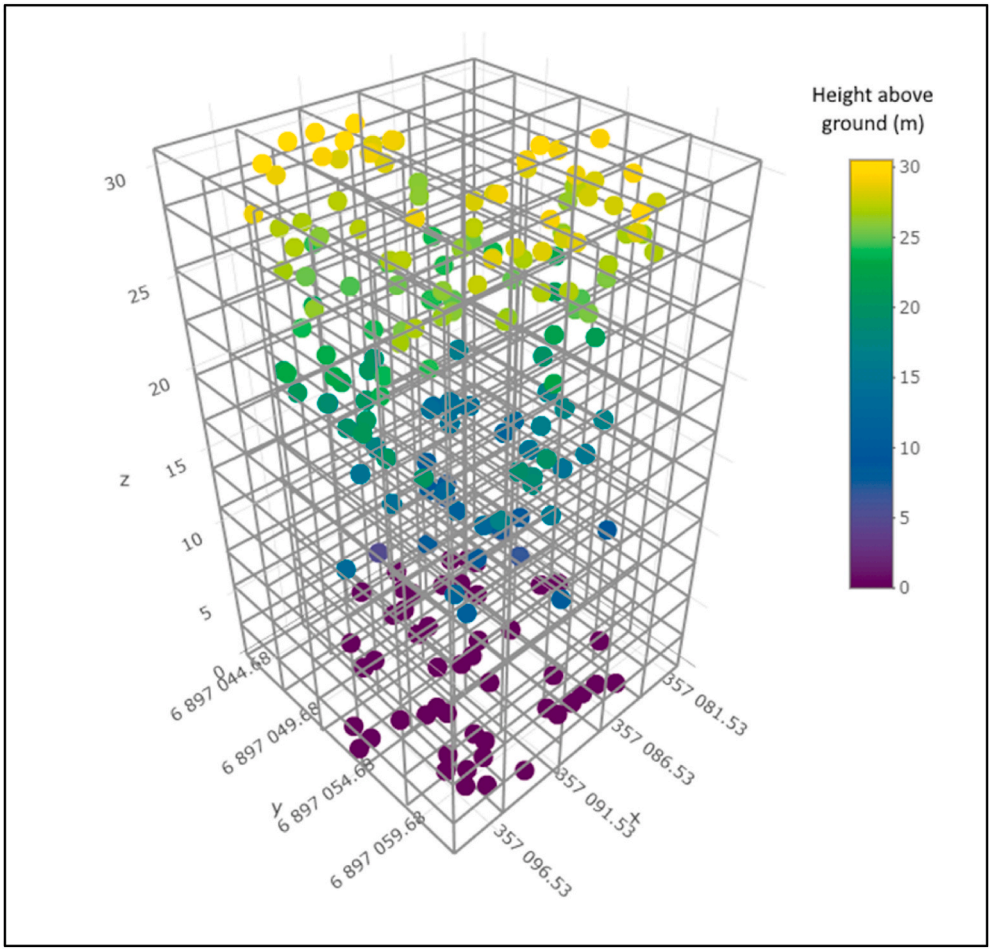



**Fig. 3.** Voxel space with laser returns.

- median of the absolute deviations from the overall median,
- inner and outer volume,
- gap area,
- rumple index.

Height metrics were calculated for first and last returns except for inner and outer volume, gap area and rumple index for which only first returns were used.

Texture features based on grey-level co-occurrence matrices were calculated with the radiomics R-package (Hall-Beyer, 2017) using first returns only.

Nearly 200 laser features were calculated and used with the k-NN and ANN methods.

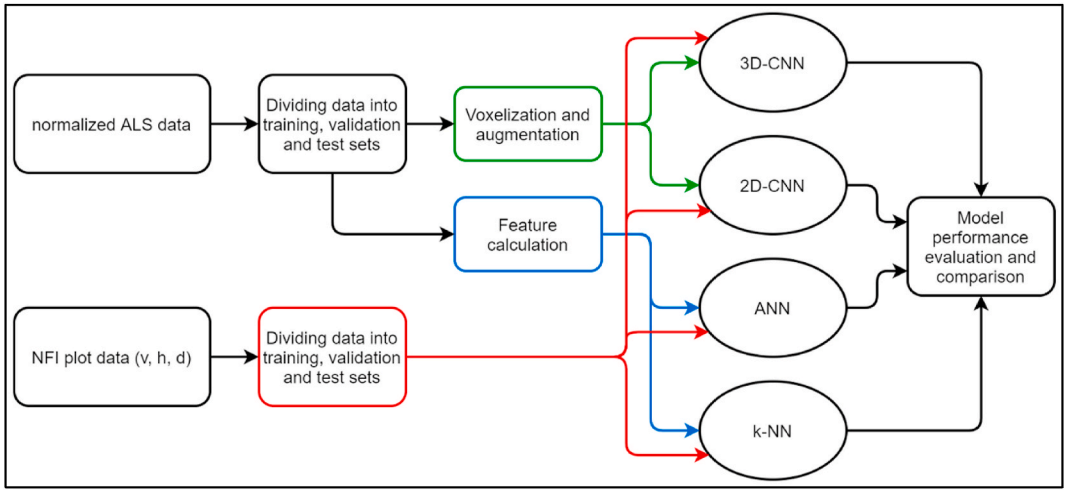



**Fig. 4.** Workflow diagram.

### 3.2. k-nearest neighbors

The k-nearest neighbors (k-NN) method was used as a benchmark for comparison with neural networks. K-NN has been successfully applied in estimating forest stand variables (e.g. Kilkki and Päivinen, 1987; Tuominen et al., 2014) and is in operational use in the national forest inventory of Finland (Tomppo, 1991).

The variables estimated were total growing stock, mean height and mean diameter. In the training of the k-NN model the training dataset with 1044 plots was used. Forest variables for sample plots were estimated as weighted averages of $k$ nearest neighbors' field observations (Eq. (1)). Nearest neighbors were selected based on Euclidean distances in a $p$-dimensional feature space where $p$ equals the number of remote sensing features in use. Inverse weighting of nearest neighbors' forest variables was performed using Euclidean distances (Eq. (2)) to reduce the bias (Altman, 1992) and averaging of estimates at the higher end of dependent variables' distributions, a known shortcoming of the k-NN method (Magnussen et al., 2010).

$$\hat{y} = \sum_{i=1}^{k} w_i y_i \tag{1}$$

where: $\hat{y}$ = estimate for variable $y$, $k$ = number of nearest neighbors, $y_i$ = field observation of variable $y$ of the $i$th nearest neighbor, weight of the $i$th nearest neighbor (Eq. (2)):

$$w_i = \frac{1}{d_i^g} \Big/ \sum_{i=1}^{k} \frac{1}{d_i^g} \tag{2}$$

where: $d_i$ = Euclidean distance of the $i$th nearest neighbor, $g$ = distance weighting parameter.

Due to high dimensionality of the feature space a genetic algorithm called *genalg* implemented in the R language (Willighagen and Ballings, 2015) was used to select and weight a suitable set of features for the k-NN prediction of forest variables. The same algorithm was also used to optimize $k$ and $g$. An initial set of random combinations of features was fed into the algorithm. Over a pre-set number of iterations the feature combinations were evolving. Features were added or removed from the initial combinations to minimize a predefined evaluation value. For the k-NN based predictions, the evaluation value (EV) (Eq. (3)) was the sum of weighted root mean square errors (RMSE) (Eq. (4)) and absolute values of unweighted biases (Eq. (5)) of estimated variables calculated by applying the leave-one-out cross-validation method. This formulation gives more weight to the bias compared to using merely RMSE ($RMSE = \sqrt{(var + bias^2)}$), which was considered as important for a large area inventory. Forest variables were estimated for each sample plot using the rest of the reference dataset and the features selected in each feature combination. Estimates were then compared to field observations and the fitness of feature combinations was evaluated by means of the evaluation value.

$$EV = \sum_{j=1}^{J} \left( w_j RMSE_j + \left| bias_j \right| \right) \tag{3}$$

$$RMSE_j = \sqrt{\frac{\sum_{i=1}^{n} \left( \hat{y}_{ji} - y_{ji} \right)^2}{n}} \tag{4}$$

$$bias_j = \frac{\sum_{i=1}^{n} \left( \hat{y}_{ji} - y_{ji} \right)}{n} \tag{5}$$

where: $w_j$ = weight of RMSE of variable $j$, $RMSE_j$ = RMSE of variable $j$, $bias_j$ = bias of variable $j$, $\hat{y}_{ji}$ = estimate for variable $j$ and sample plot $i$, $y_{ji}$ = field observation of variable $j$ and sample plot $i$, $n$ = number of sample plots in the training set.

Based on the *genalg* procedure the weights applied to RMSE values were 0.6, 0.2 and 0.2 for total growing stock, mean height and mean diameter respectively. RMSE weights set the importance of estimated variables in the training process. In the final k-NN model number of neighbors ($k$) of 6 and distance weighting parameter (g) of 2.0 was used.

### 3.3. Neural networks

#### 3.3.1. Artificial neural network

The ANN used in this study was implemented in R statistical software using Keras with TensorFlow backend. The ANN had a single hidden layer with $N$ neurons. The number of neurons in the hidden layer was treated as a hyper-parameter and the performance was tested with values 3–6, 10, 20, and 30. There are many ways to define the number of neurons in the hidden layer but experts (e.g. Heaton, 2008) generally suggest trial and error as the best method. Laser features used in the k-NN method served as independent variables for the ANN. Additionally to the training set of 1044 sample plots validation set was used to avoid overfitting.

Training was carried out in a predefined number of epochs (training cycle). In each epoch the network was trained by iterating through the training dataset and feeding batches of 25 sample plots at the time for faster and more memory-efficient training. It has been also reported that the use of small batch sizes enhances the generalizability of the model (Keskar et al., 2017). The loss function used to evaluate the model performance after each training step was a weighted mean (Eq. (6)) of

mean squared errors (MSE, Eq. (7)) suitable for regression problems. The variables were weighted the same way as in the k-NN method. After each of the 200 training epochs model performance was also evaluated by calculating the MSE on the estimated variables of the validation set. The validation MSE was monitored, and training was stopped if it was not improving after 20 epochs. This practice called early stopping, similarly to the use of validation data, aims to avoid overfitting. The model was saved after each epoch, if the validation MSE indicated that the current model outperformed the previous one.

$$EV = \sum_{j=1}^{J} w_j MSE_j \tag{6}$$

$$MSE_j = \frac{\sum_{i=1}^{n} \left(\hat{y}_{ji} - y_{ji}\right)^2}{n} \tag{7}$$

where: $\hat{y}_i$ = estimate for variable $j$ and sample plot $i$, $y_i$ = field observation of variable $j$ and sample plot $i$, $n$ = number of sample plots in the validation set (225).

### 3.3.2. 3D convolutional neural network

The three-dimensional CNN based on Google's Inception V3 (Szegedy et al., 2016) used in this study was implemented by Ayrey and Hayes (2018) in Python using Google's TensorFlow library. The CNN was created in a way that three-dimensional convolutional kernels would scan the voxel spaces unlike the original implementation for two-dimensional pixels. The network consists of five initial convolutional layers with two pooling layers and eleven inception layers. Convolutional layers are a series of moving windows that aim at extracting spatial features from the input data. Pooling layers are used between convolution layers to extract the most important information from the feature maps created by convolution layers and reduce dimensionality. Thus, pooling reduces the number of parameters to learn and the amount of computation. Inception layers or inception modules are groups of convolutions with the ability to recognize various features by applying different convolutional kernel sizes at the same level of the network. After each convolution batch normalization is carried out. The outputs of convolutions are normalized to have mean zero and standard deviation one. Normalization is carried out over batches of training data fed to the network in each training epoch. Batch normalization is said to enable faster and more stable training of NNs (Ioffe and Szegedy, 2015). Batch normalization is followed by an activation layer using either exponential or rectified linear unit activation function (ELU, ReLU). The activation function takes the output from one layer in the network and transforms it before the information is passed on to the next layer. Activation functions typically add non-linearity to NNs helping to detect complex patterns in the data. Fig. 5 shows the simplified architecture of the 3D-CNN.

Low-level coding of the network enabled the implementation of some additional features. The original code was edited to add the possibility to estimate multiple forest variables at the same time, enable alternate training (Godwin, 2016) and augmentation on-the-fly. Alternate training means that in each training epoch the network is trained based on one dependent variable at the time, iterating through all dependent variables in the data. The opposite of alternate training is when all dependent variables are estimated at the same time. On-the-fly augmentation doubles the size of the batch of training data as described in section 2.2 ensuring that the network is trained in each epoch using original and augmented samples. MSE was used as the loss function for training the network. The number of epochs was set to 20 000 to avoid interrupting the training process in case the loss function temporarily plateaus (i.e., gets stuck in a local minimum). To avoid overfitting the best performing model was saved based on the same criteria as described in 3.3.1.

Additionally, to voxel types described in section 2.2, multiple options for several hyper-parameters typically used in NNs were tested. Learning

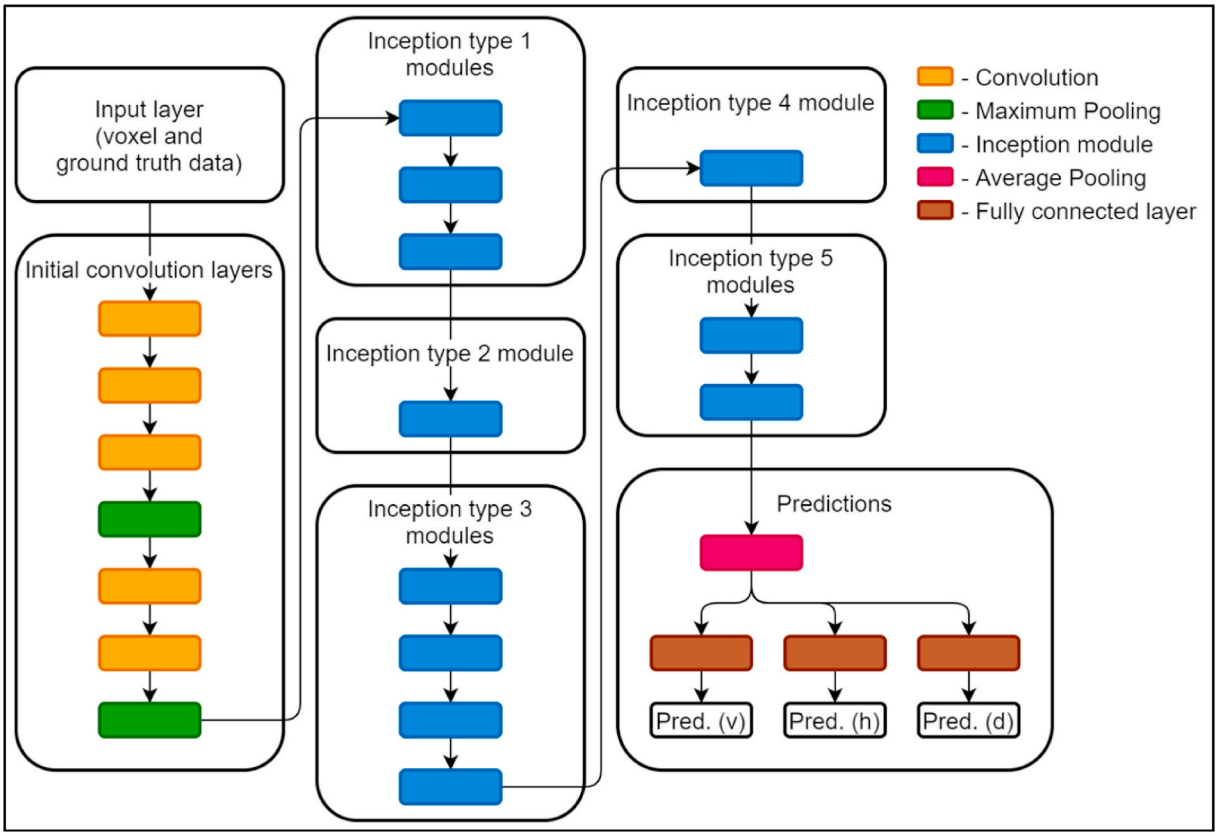



**Fig. 5.** Architecture of the 3D-CNN.

rate defines how much the network is changed after each training step (Goodfellow et al., 2016). Values tested for learning rate were: 0.01, 0.001, 0.0001 and 0.00001. Batch sizes tested were 10, 16 and 32 including augmented samples due to on-the-fly augmentation. Because of the complexity of CNNs and the time required for training, only one randomly selected batch of training data was used in each epoch. Based on preliminary testing binary voxel type, learning rate of 0.0001 and batch size of 16 was chosen.

#### 3.3.3. 2D convolutional neural network

For this study a two-dimensional CNN was built in Python using Keras with TensorFlow backend. The 2D-CNN was based on AlexNet (Krizhevsky et al., 2017), originally designed for the ImageNet Large-Scale Visual Recognition Challenge. The network has five convolutional layers, with the first, second and fifth followed by pooling layers and finally three fully connected layers. Each convolution is followed by ReLU activation and batch normalization. The architecture of the 2D-CNN is considerably simpler, and its training is faster and more memory-efficient than 3D-CNN. 2D-CNN treats each voxel layer as two-dimensional, rasterized data. Unlike for 3D-CNN, learning rate was implemented as a scheduled parameter which was decreasing according to a polynomial function as training was progressing. The loss function and model saving were implemented in the same way as in the previous architectures. Voxel types and batch sizes of 10, 16 and 32 were tested. Normalized voxel type and a batch size of 32 was selected to be used with 2D-CNN. Augmentation on-the-fly was not possible due to high-level functions in the code, hence pre-augmented data was used for training. Fig. 6 shows the architecture of the 2D-CNN.

The input data and shell scripts are available at: https://github.com/balazsan/ALS_NNs (see also Supplementary materials).

### 3.4. Evaluation and comparison of model performances

All models tested in this study have a large number of parameters whose values are defined randomly in the beginning of the training process. This leads to slightly or sometimes significantly different results when model training is carried out multiple times. To account for this issue each model with each parameter combination was run ten times parallelly and the best preforming results were selected.

To compare the models forest variables were estimated for the test set of 225 sample plots. This dataset was not used at any point of the training process allowing the best comparability of results. Model performances were evaluated in terms of relative RMSE (RMSE %), relative bias (bias %) and coefficient of determination ($R^2$) which were calculated based on estimations and field observations.

$$RMSE\ \% = 100 \cdot \frac{RMSE}{\bar{y}} \tag{8}$$

$$bias\ \% = 100 \cdot \frac{bias}{\bar{y}} \tag{9}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{n} (y_i - \bar{y})^2} \tag{10}$$

where: $\hat{y}_i$ = estimate for variable $y$ and sample plot $i$, $y_i$ = field observation of variable $y$ and sample plot $i$, $\bar{y}$ = mean of field observations, $n$ = number of sample plots in the test set (225).

## 4. Results

Model performances were compared based on the accuracy scores of predictions in the test dataset with 225 sample plots. Estimation accuracy results are presented in Fig. 7. In terms of RMSE% 3D-CNN outperformed the k-NN method in estimating all three forest stand variables. Estimation accuracies were 1.31%, 0.98% and 1.23% lower for total growing stock, mean height and mean diameter respectively. Relative biases were one magnitude lower for mean height and diameter and the same magnitude for total growing stock as the benchmark method. Accuracies achieved with the ANN model were slightly better than the k-NN method's marks for all three dependent variables. ANN's mean diameter estimates resulted in the lowest RMSE%, only 0.29%

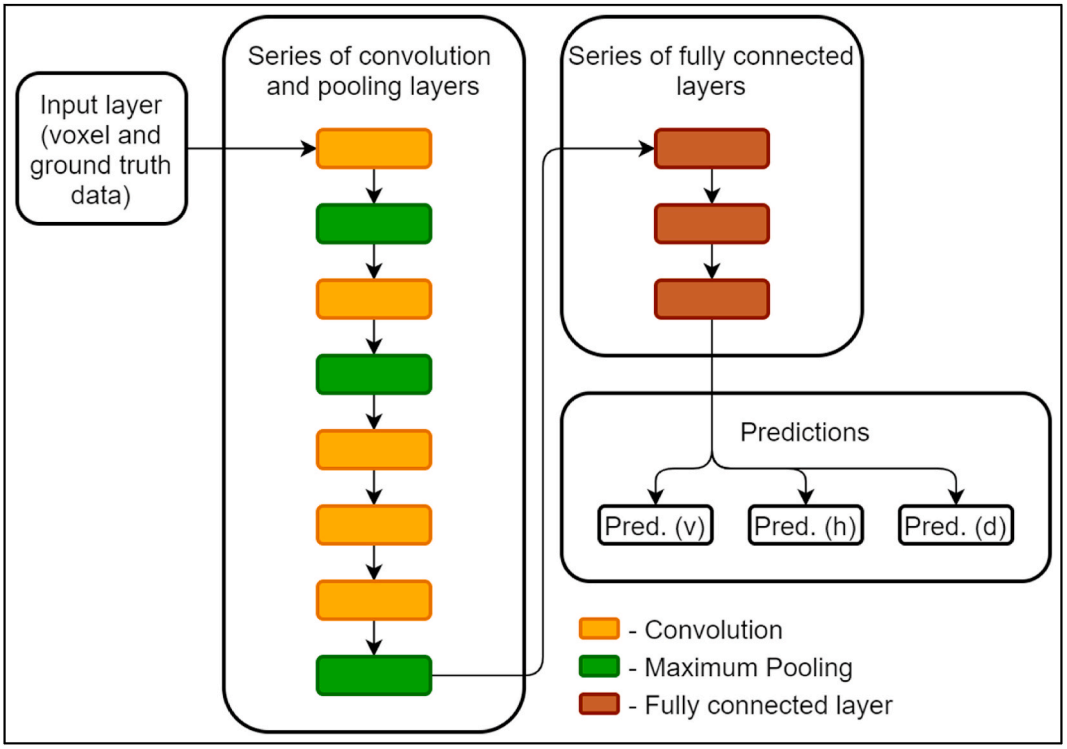



**Fig. 6.** Architecture of the 2D-CNN.

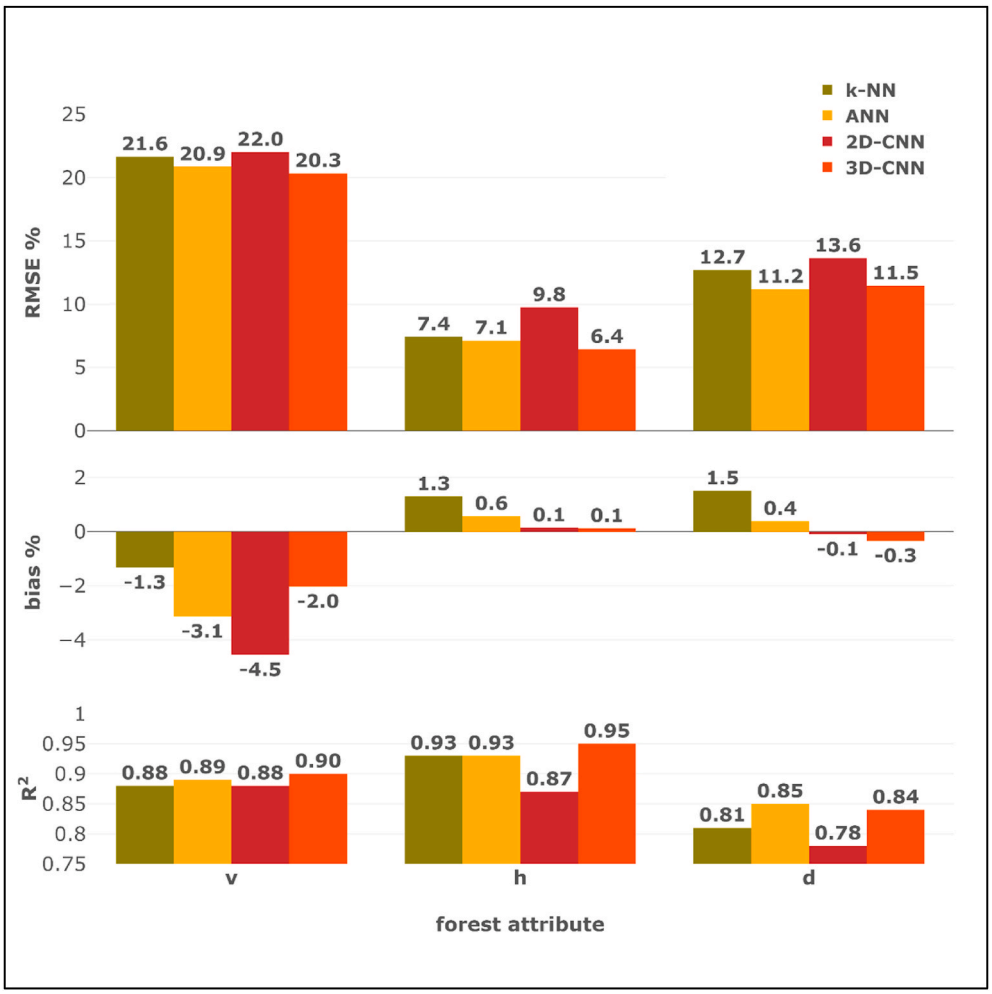



**Fig. 7.** Summary of model performances.

lower than the same indicator of the 3D-CNN. 2D-CNN performance was the poorest in terms of RMSE%. For a sake of comparison, the best 5-predictor linear models were also estimated for all the variables (Appendix 1a). Linear models perform comparably to k-NN.

Based on the predicted vs. observed forest attribute scatterplots shown in Fig. 8 it is obvious that all models tend to underestimate forest variables near the higher end of their distributions. However, underestimation is slightly smaller with NNs. Height estimates of ANN and 3D-CNN are visibly closer to the identity line than the other models' estimates. In order to assess the importance of the differences between k-NN and other methods, we provide Bland-Altman plots of the predictions (see Appendix 2) (Bland and Altman, 1995). The main difference between k-NN and neural network methods is that k-NN predictions have a smaller variance.

Scatter plots with covariances of all response variable combinations and all tested methods are in Appendix 3. The covariances of mean height and mean diameter shown in Fig. 10 were for all methods lower than for the observations. Closest to the observed co-variation was ANN. 2D CNN reduced the covariance more than the other methods (see Figs. 10–12).

## 5. Discussion

Based on our results it was found that improvement could be achieved in estimating forest stand variables by using a 3D-CNN compared to the k-NN benchmark method. It is, however, important to point out that differences between the methods tested are small.

One obvious reason for a moderate difference to the advantage of NNs is the relatively low sample size. Whereas an extremely high number of samples is suggested (Ayrey and Hayes, 2018; Benkendorf and Hawkins, 2020) to be necessary for a deep neural network to train successfully, reference data with thousands of samples plots are rarely available and the requirement of high numbers of reference plots is often poorly compatible with the requirements of practical forest inventories based on airborne laser scanning. Using methods such as k-NN, typically 300–700 sample plots are used per inventory area, when comprehensive set of inventory variables for forest management need to be predicted, whereas significantly smaller number of sample plots (even in the class of 50–100) have been found sufficient when estimating variables such as volume of growing stock (Gobakken and Næsset, 2008; Gobakken et al., 2013; Tuominen et al., 2014; Tomppo et al., 2016; Stereńczak et al., 2018). With NNs, there are no guidelines to a minimum sample size, recommendations mostly being "the more the better".

Ayrey and Hayes (2018) reported 11% decrease of RMSE in estimating biomass by the same 3D-CNN compared to results obtained by a Random Forest model trained with traditional height metrics. The ALS data used in their study was ranging from 9 to 15 points/$m^2$ as opposed to 0.91 point/$m^2$ of the data available to the present study. It seems that CNNs were not able to extract significantly better features from the sparse point cloud than the calculated features that were in use for k-NN and ANN. Low point density might also explain why binary was more suitable than count-of-points voxel type to be used with the 3D-CNN.

Starting from 2020 a new airborne laser scanning project has been launched in Finland, aiming at covering the entire country by ALS data

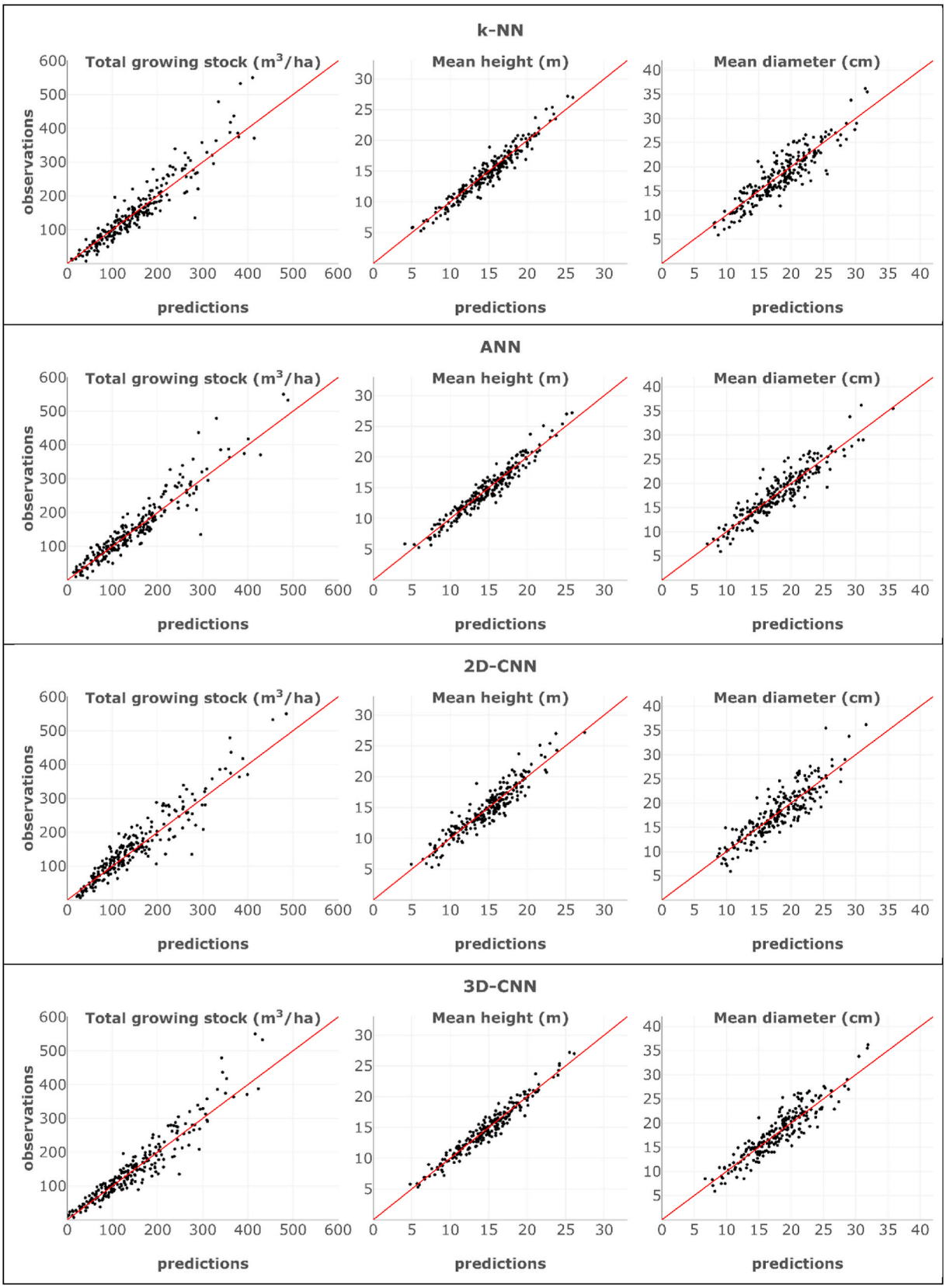



**Fig. 8.** Scatterplots of predicted vs. observed forest attributes and identity lines in red. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

with average point density 5 points/m$^2$ six-year inventory cycle, thus markedly improving the spatial and temporal resolution of open access lidar data for forestry. This denser laser data could be used in verifying if higher point density can significantly enhance the performance of 3D-CNN with similar sized reference dataset. A proficient interpreter should be able to utilize significantly wider assortment of remote sensing features than typically applied in ALS-based forest inventories, such object shape, pattern, and association with other objects. The extraction of these features is not possible by straightforward algorithms, and assuming that the CNNs are able to imitate the interpreter's perception, higher resolution of ALS point clouds should allow better recognition of forest characteristics by CNNs, thus adding their advantage over other machine learning methods.

In this study, we tested the methods for the basic forest variables, volume, mean height and mean diameter, which are in general the variables that can be estimated with highest accuracy. It is possible that the minor differences are due to the tested variables, and the benefits from CNN may be higher with more challenging variables, for instance species-specific volumes (Packalén and Maltamo, 2007; Tuominen et al., 2017). In such conditions, new features possibly obtainable from convolutions could provide additional information. This needs to be studied in the future.

In 2D-CNN convolutional kernels are scanning each voxel layer in vertical direction individually much like individual channels of an image. This technique fails to extract features across voxel layers which could be the reason for the weakest performance of models tested in this study. Thus, the vertical distribution of the point cloud involves information not achievable from analyzing the layers separately. In a 2D approach, such information can be obtained by forming higher-level features from the separate layers. It is possible that such higher-level features would require additional hidden layers.

Based on different studies, the best performing machine learning method seems to be somewhat dependent on the data set used for training, due to factors such as the variation of the inventory variables and the complexity of the remote sensing data (i.e., high dimensionality of the feature space and the mutual correlation between the features). ANN has shown some potential in slightly more accurate estimates compared to k-NN but marked improvements might require a bigger training sample. One advantage of ANN could be more accurate estimates towards the higher end of forest variable distributions. However, the best methods vary depending on the conditions of the tests. For instance, Shah et al. (2020) tested CNN and Random Forest method in estimating canopy height by optical satellite imagery, resulting in somewhat better estimates with Random Forest, although they point out the limitations of that method in the extraction and utilization of satellite image features. Dalla Corte et al. (2020) have tested several machine learning methods including Support vector regression, Random Forest, ANN, and Extreme gradient boosting with high density UAV lidar point cloud data and based on their results the Support vector regression had the best performance in predicting individual tree attributes. In this study, the performance of CNN was not markedly superior to other methods, as relatively sparse lidar pulse density was used. When higher pulse density lidar data becomes more widely available, the advantage of CNN is likely to become more prominent in predicting forest attributes and in recognizing structural forest characteristics.

## 6. Conclusions

In the present study we have shown that neural networks, either used with traditional ALS features or with voxelized ALS point clouds, are able to slightly outperform the benchmark k-NN method combined with a genetic algorithm in estimating forest stand variables. Differences in model performances were small and further investigation is necessary to confirm if higher ALS point density could yield in better performance of CNNs with a relatively small training dataset.

## Acknowledgments

This work was supported by the Natural Resources Institute Finland and by the Academy flagship UNITE (decision number 337655). The authors would like to thank Elias Ayrey for providing the 3D-CNN and support for the adaptation of the network for this study. The authors also wish to acknowledge CSC – IT Center for Science, Finland, for computational resources.

## Appendix 1a: Performance of the linear models

Fitted models are all of the form. $\log(y) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_4 + \beta_5 x_5 + \varepsilon$

The selected variables for Volume were: height where 95% of ALS returns have accumulated (first echoes), intensity mean value (last echoes), gap area, canopy surface model difference entropy, and canopy surface model homogeneity.

In the training data and logarithmic scale $R^2 = 94.11\%$ and RMSE = 0.2151.

In test data and arithmetic scale bias% = 1.043523, RMSE% = 21.9298 and $R^2 = 88.29818$.

The selected variables for Height were: percentage of ALS canopy points having height greater than 95th height percentile (first echoes), percentage of canopy points having height greater than 90th height percentile (first echoes), percentage of canopy points having height greater than 60th height percentile (first echoes), canopy surface model inverse difference moment, cubic mean (first echoes).

In the training data and logarithmic scale $R^2 = 94.18\%$ and RMSE = 0.0755.

In test data and arithmetic scale bias% = 1.181472, RMSE% = 6.938457 and $R^2 = 93.62741$.

The selected variables for Diameter were: percentage of canopy points having height greater than 20th height percentile (last echoes), height where 95% of ALS returns have accumulated (first echoes), canopy surface model difference Entropy, canopy surface model energy, ALS intensity canopy shade.

In the training data and logarithmic scale $R^2 = 82.69\%$and RMSE 0.1331.

In test data and arithmetic scale bias% = −0.4696578, RMSE% = 12.09116 and $R^2 = 82.403$.

## Appendix 1. b: Features selected for k-NN predictions

Selected variables for the k-NN predictions were:

- percentage of ALS canopy points having height greater than 5th height percentile (first echoes)
- height where 20% ALS returns have accumulated (last echoes)
- height where 60% of returns have accumulated (last echoes)
- height where 95% of ALS returns have accumulated (first echoes)

- cumulative percentage of returns in the 4th layer (of canopy layers 1–5)
- ratio of the number of vegetation points to the number of ground points
- median of the absolute deviations from the overall median
- mean of ALS intensity (first echoes)
- percentage of ALS canopy points having height greater than 5th height percentile (last echoes)
- height where 75% of ALS canopy points have accumulated (last echoes)
- height where 85% of ALS canopy points have accumulated (last echoes)
- cumulative percentage of returns in the 3rd layer (of canopy layers 1–5)
- inner volume
- gap area
- canopy surface model sum variance

## Appendix 2. Bland-Altman plots

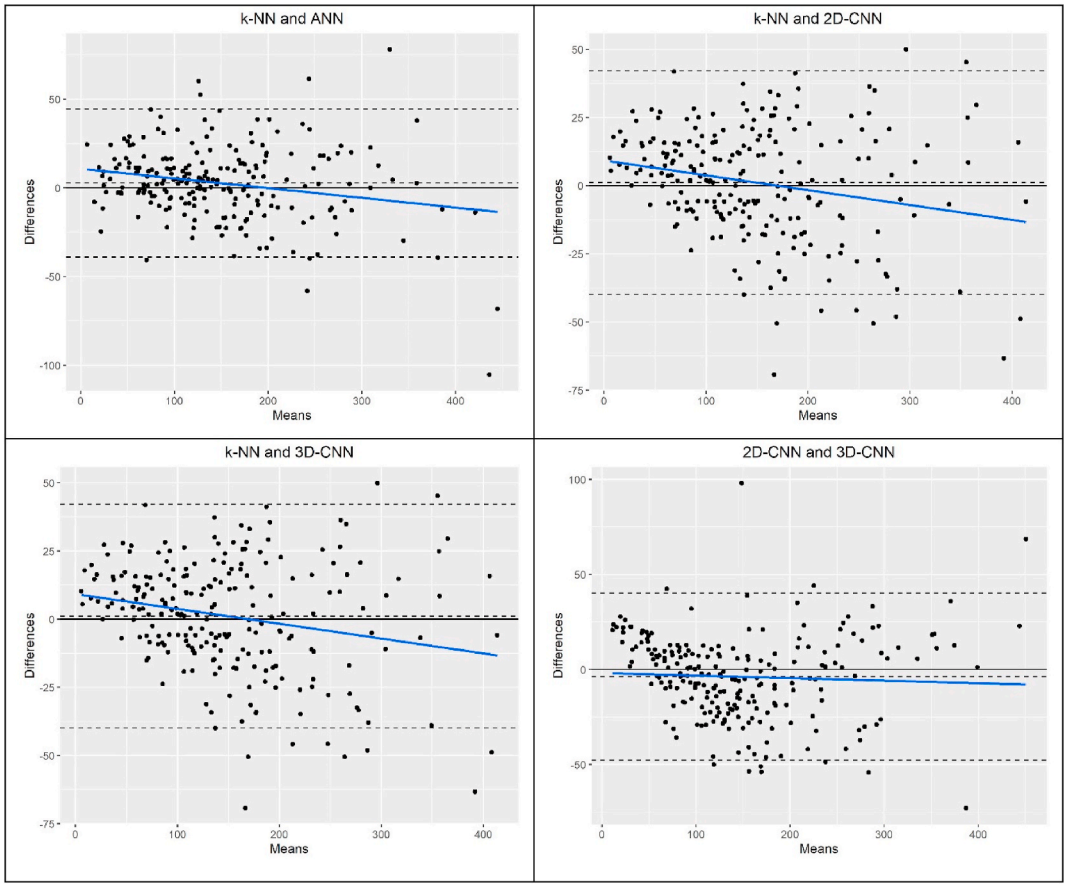



**Fig. 9.** The difference between predictions from two method pairs compared to the mean of the predictions. The dashed lines are 95% confidence interval. The dashed line in the middle is bias, depicting the average difference between the methods. The blue trend line depicts the difference in the variances of the predictions. (Downward trend means the first and upward trend the second method has smaller variance).

## Appendix 3. Covariances of all response variable combinations

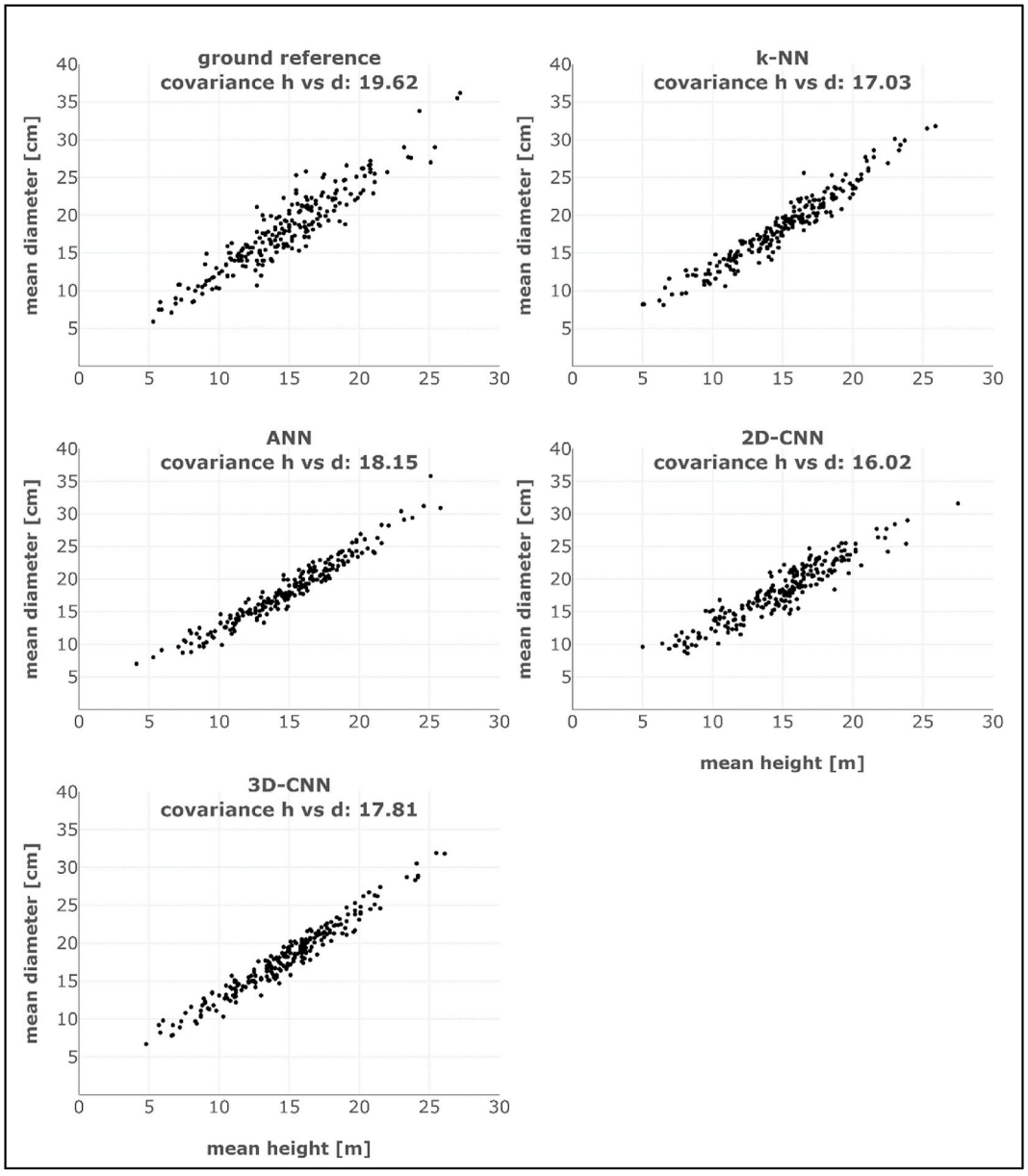



**Fig. 10.** Scatterplots showing mean height vs. mean diameter and covariances for field observations and predictions.

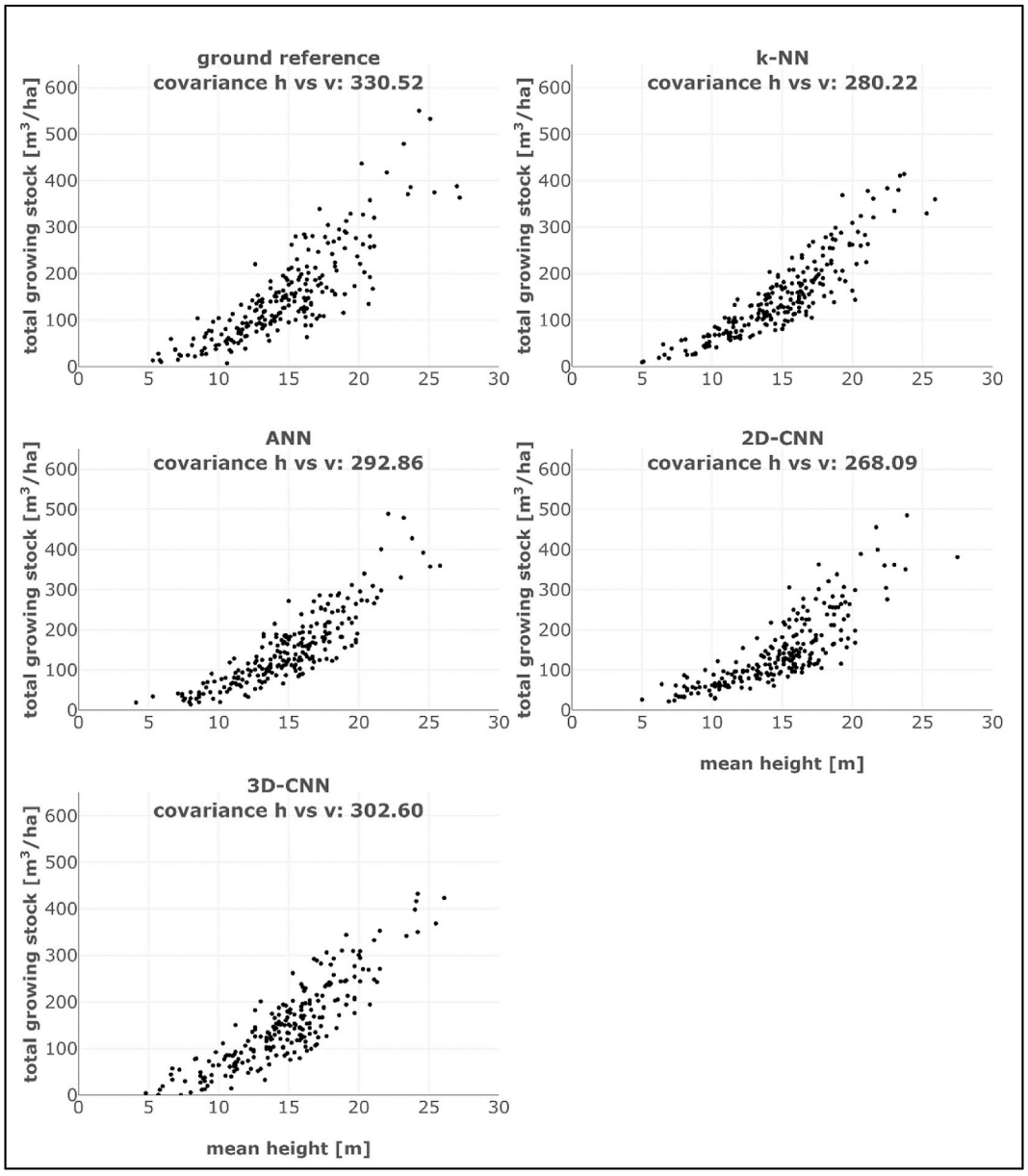



**Fig. 11.** Scatterplots showing mean height vs. volume of total growing stock and covariances for field observations and predictions.

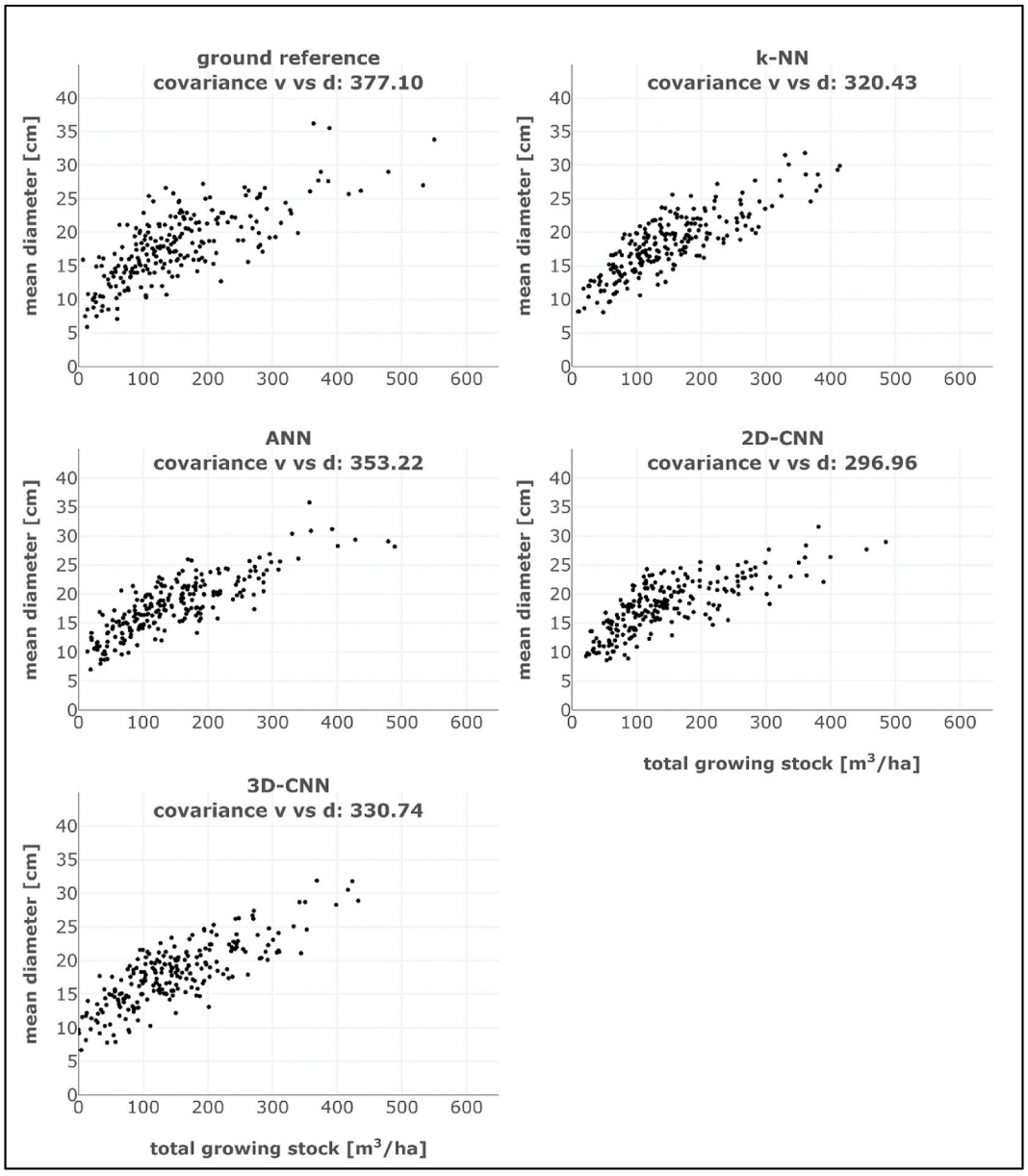



**Fig. 12.** Scatterplots showing volume of total growing stock vs. mean diameter and covariances for field observations and predictions.

## References


Altman, N.S., 1992. An introduction to kernel and nearest-neighbor nonparametric regression. Am. Statistician 46 (3), 175–185. https://doi.org/10.1080/00031305.1992.10475879.

Ayrey, E., Hayes, D.J., 2018. The use of three-dimensional convolutional neural networks to interpret LiDAR for forest inventory. Rem. Sens. 10 (4), 649. https://doi.org/10.3390/rs10040649.

Benkendorf, D.J., Hawkins, C.P., 2020. Effects of sample size and network depth on a deep learning approach to species distribution modeling. Ecol. Inf. 60, 101137. https://doi.org/10.1016/j.ecoinf.2020.101137.

Bishop, C.M., 1996. Pre-processing and post-processing. In: Neural Networks for Pattern Recognition, first ed. Clarendon Press, Oxford : New York, p. 296.

Bland, J.M., Altman, D.G., 1995. Comparing methods of measurement: why plotting difference against standard method is misleading. Lancet (London, England) 346 (8982), 1085–1087. https://doi.org/10.1016/s0140-6736(95)91748-9.

Corte, A.P.D., Souza, D.V., Rex, F.E., Sanquetta, C.R., Mohan, M., Silva, C.A., Broadbent, E.N., 2020. Forest inventory with high-density UAV-Lidar: machine learning approaches for predicting individual tree attributes. Comput. Electron. Agric. 179, 105815. https://doi.org/10.1016/j.compag.2020.105815.

Fricker, G.A., Ventura, J.D., Wolf, J.A., North, M.P., Davis, F.W., Franklin, J., 2019. A convolutional neural network classifier identifies tree species in mixed-conifer forest from hyperspectral imagery. Rem. Sens. 11 (19), 2326. https://doi.org/10.3390/rs11192326.

Gobakken, T., Korhonen, L., Næsset, E., 2013. Laser-assisted selection of field plots for an area-based forest inventory. Silva Fenn. 47 (5) https://doi.org/10.14214/sf.943.

Gobakken, T., Næsset, E., 2008. Assessing effects of laser point density, ground sampling intensity, and field sample plot size on biophysical stand properties derived from airborne laser scanner data. Can. J. For. Res. 38 (5), 1095–1109. https://doi.org/10.1139/X07-219.

Godwin, J., 2016. Multi-task learning in tensorflow (Part 1). June 30, Retrieved August 4, 2021, from. https://jg8610.github.io/Multi-Task/.

Goodfellow, I., Bengio, Y., Courville, A., 2016. Gradient-based optimization. In: Deep Learning, Illustrated edition. The MIT Press, Cambridge, Massachusetts, p. 86.

Hall-Beyer, M., 2017. GLCM Texture: A Tutorial V. 3.0 March 2017.

Hamraz, H., Jacobs, N., Contreras, M., Clark, C., 2018. Deep learning for conifer/deciduous classification of airborne LiDAR 3D point clouds representing individual trees. ISPRS J. Photogrammetry Remote Sens. 158 https://doi.org/10.1016/j.isprsjprs.2019.10.011.

Heaton, J., 2008. Introduction to Neural Networks for Java, second ed. Heaton Research, Inc.

Illarionova, S., Trekin, A., Ignatiev, V., Oseledets, I., 2021. Neural-based hierarchical approach for detailed dominant forest species classification by multispectral satellite imagery. IEEE J. Sel. Top. Appl. Earth Obs. Rem. Sens. 14, 1810–1820. https://doi.org/10.1109/JSTARS.2020.3048372.

Ioffe, S., Szegedy, C., 2015. Batch normalization: accelerating deep network training by reducing internal covariate shift. ArXiv:1502.03167 [Cs]. Retrieved from. http://arxiv.org/abs/1502.03167.

Jensen, R.R., Binford, M.W., 2004. Measurement and comparison of Leaf Area Index estimators derived from satellite remote sensing techniques. Int. J. Rem. Sens. 25 (20), 4251–4265. https://doi.org/10.1080/01431160410001680400.

Keskar, N.S., Mudigere, D., Nocedal, J., Smelyanskiy, M., Tang, P.T.P., 2017. On large-batch training for deep learning: generalization gap and sharp minima. ArXiv: 1609.04836 [Cs, Math]. Retrieved from. http://arxiv.org/abs/1609.04836.

Kilkki, P., Päivinen, R., 1987. Reference Sample Plots to Combine Field Measurements and Satellite Data in Forest Inventory.

Kirchhoefer, M., Schumacher, J., Adler, P., Kändler, G., 2017. Considerations towards a novel approach for integrating angle-count sampling data in remote sensing based forest inventories. Forests 8 (7), 239. https://doi.org/10.3390/f8070239.

Krizhevsky, A., Sutskever, I., Hinton, G.E., 2017. ImageNet classification with deep convolutional neural networks. Commun. ACM 60 (6), 84–90. https://doi.org/10.1145/3065386.

Lefsky, M.A., Cohen, W.B., Harding, D.J., Parker, G.G., Acker, S.A., Gower, S.T., 2002. Lidar remote sensing of above-ground biomass in three biomes. Global Ecol. Biogeogr. 11 (5), 393–399. https://doi.org/10.1046/j.1466-822x.2002.00303.x.

Li, Z., Khananian, A., Fraser, R.H., Cihlar, J., 2001. Automatic detection of fire smoke using artificial neural networks and threshold approaches applied to AVHRR imagery. IEEE Trans. Geosci. Rem. Sens. 39 (9), 1859–1870. https://doi.org/10.1109/36.951076.

Liu, T., Sun, Y., Wang, C., Zhang, Y., Qiu, Z., Gong, W., Duan, X., 2021. Unmanned aerial vehicle and artificial intelligence revolutionizing efficient and precision sustainable forest management. J. Clean. Prod. 311, 127546. https://doi.org/10.1016/j.jclepro.2021.127546.

Magnussen, S., Tomppo, E., McRoberts, R.E., 2010. A model-assisted k-nearest neighbour approach to remove extrapolation bias. Scand. J. For. Res. 25 (2), 174–184. https://doi.org/10.1080/02827581003667348.

Maltamo, M., Eerikäinen, K., Pitkänen, J., Hyyppä, J., Vehmas, M., 2004. Estimation of timber volume and stem density based on scanning laser altimetry and expected tree size distribution functions. Remote Sens. Environ. 90 (3), 319–330. https://doi.org/10.1016/j.rse.2004.01.006.

Moura, M.M., de Oliveira, L.E.S., Sanquetta, C.R., Bastos, A., Mohan, M., Corte, A.P.D., 2021. Towards amazon forest restoration: automatic detection of species from UAV imagery. Rem. Sens. 13 (13), 2627. https://doi.org/10.3390/rs13132627.

Næsset, E., 2002. Predicting forest stand characteristics with airborne scanning laser using a practical two-stage procedure and field data. Remote Sens. Environ. 80 (1), 88–99. https://doi.org/10.1016/S0034-4257(01)00290-5.

Næsset, E., 2004. Practical large-scale forest stand inventory using a small-footprint airborne scanning laser. Scand. J. For. Res. 19 (2), 164–179. https://doi.org/10.1080/02827580310019257.

Packalén, P., Maltamo, M., 2006. Predicting the plot volume by tree species using airborne laser scanning and aerial photographs. For. Sci. 52 (6), 611–622.

Packalén, P., Maltamo, M., 2007. The k-MSN method for the prediction of species-specific stand attributes using airborne laser scanning and aerial photographs. Remote Sens. Environ. 109 (3), 328–341. https://doi.org/10.1016/j.rse.2007.01.005.

Packalén, P., Suvanto, A., Maltamo, M., 2009. A two stage method to estimate species-specific growing stock. Photogramm. Eng. Rem. Sens. 75 (12), 1451–1460. https://doi.org/10.14358/PERS.75.12.1451.

Pohjankukka, J., Tuominen, S., Pitkänen, J., Pahikkala, T., Heikkonen, J., 2018. Comparison of estimators and feature selection procedures in forest inventory based on airborne laser scanning and digital aerial imagery. Scand. J. For. Res. 33 (7), 681–694. https://doi.org/10.1080/02827581.2018.1482955.

Roussel, J.-R., 2021. Stdmetrics · Jean-Romain/lidR Wiki. Retrieved July 30, 2021, from GitHub website: https://github.com/Jean-Romain/lidR.

Shah, S.A.A., Manzoor, M.A., Bais, A., 2020. Canopy height estimation at landsat resolution using convolutional neural networks. Machine Learning and Knowledge Extraction 2 (1), 23–36. https://doi.org/10.3390/make2010003.

Shorten, C., Khoshgoftaar, T.M., 2019. A survey on image data augmentation for deep learning. Journal of Big Data 6 (1), 60. https://doi.org/10.1186/s40537-019-0197-0.

Shukla, N., Fricklas, K., 2018. Machine Learning with TensorFlow. Manning, Shelter Island, NY.

Stereńczak, K., Lisanczuk, M., Parkitna, K., Mitelsztedt, K., Mroczek, P., Miścicki, S., 2018. The influence of number and size of sample plots on modelling growing stock volume based on airborne laser scanning. https://doi.org/10.12841/WOOD.1644-3985.D11.04.

St-Onge, B., Vega, C., Fournier, R.A., Hu, Y., 2008. Mapping canopy height using a combination of digital stereo-photogrammetry and lidar. Int. J. Rem. Sens. 29 (11), 3343–3364. https://doi.org/10.1080/01431160701469040.

Syrris, V., Hasenohr, P., Delipetrev, B., Kotsev, A., Kempeneers, P., Soille, P., 2019. Evaluation of the potential of convolutional neural networks and random forests for multi-class segmentation of sentinel-2 imagery. Rem. Sens. 11 (8), 907. https://doi.org/10.3390/rs11080907.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z., 2016. Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2818–2826. Retrieved from. http://arxiv.org/abs/1512.00567.

Tavares Júnior, I. da S., Rocha, J. E. C. da, Ebling, Â.A., Chaves, A. de S., Zanuncio, J.C., Farias, A.A., Leite, H.G., 2019. Artificial neural networks and linear regression reduce sample intensity to predict the commercial volume of Eucalyptus clones. Forests 10 (3), 268. https://doi.org/10.3390/f10030268.

Tomppo, E., 1991. Satellite image-based national forest inventory of Finland. International Archives of Photogrammetry and Remote Sensing 28, 419–424.

Tomppo, E., Kuusinen, N., Mäkisara, K., Katila, M., McRoberts, R.E., 2016. Effects of field plot configurations on the uncertainties of ALS-assisted forest resource estimates. Scand. J. For. Res. 1–13. https://doi.org/10.1080/02827581.2016.1259425.

Tuominen, S., Pitkänen, J., Balazs, A., Korhonen, K.T., Hyvönen, P., Muinonen, E., 2014. NFI plots as complementary reference data in forest inventory based on airborne laser scanning and aerial photography in Finland. Silva Fenn. 48 (2), 19. https://doi.org/10.14214/sf.983.

Tuominen, S., Pitkänen, T., Balazs, A., Kangas, A., 2017. Improving Finnish multi-source national forest inventory by 3D aerial imaging. Silva Fenn. 51 (4) https://doi.org/10.14214/sf.7743.

Véga, C., Renaud, J.-P., Durrieu, S., Bouvier, M., 2016. On the interest of penetration depth, canopy area and volume metrics to improve Lidar-based models of forest parameters. Remote Sens. Environ. 175, 32–42. https://doi.org/10.1016/j.rse.2015.12.039.

Willighagen, E., Ballings, M., 2015. genalg: R based genetic algorithm. Retrieved March 2, 2017, from. https://cran.r-project.org/web/packages/genalg/index.html.