Bulletin of the *Transilvania* University of Braşov
Series II: Forestry • Wood Industry • Agricultural Food Engineering • Vol. 15(64) No. 1 – 2022
https://doi.org/10.31926/but.fwiafe.2022.15.64.1.2

CLASSYFING OPERATIONAL EVENTS IN CABLE YARDING BY A MACHINE LEARNING APPLICATION TO GNSS-COLLECTED DATA: A CASE STUDY ON GRAVITY-ASSISTED DOWNHILL YARDING

Stelian A. BORZ[1*] Marius CHEŢA[1]
Marcian BÎRDA[2] Andrea R. PROTO[3]

Abstract: *Cable yarding remains an important option in steep terrain timber harvesting, a reason for which new or improved operational efficiency models are required to support science and practice. Developed traditionally, these models are known to require many resources, a reason for which new approaches to the problem were researched lately, mainly by the use of Global Navigation Satellite System (GNSS) data, spatial and statistical inference systems. This study evaluates the possibility of using GNSS data and machine learning techniques to classify important cable yarding events in the time domain. Three classes were assumed by the study as being relevant for cable yarding operational setup, namely carriage moving in the uphill (MU) and downhill (MD) directions, as well as carriage stopped (S). Data collected by a consumer-grade GNSS unit was processed to extract some differential parameters which were coupled with GNSS motorial and geometric features to feed a Multi-Layer Perceptron Neural Network with Back propagation (MLPNNB) in a pre-evaluation phase which aimed at mining the data structure as a strategy to develop the best MLPNNB configuration for training and testing. Leg distance, difference in elevation, speed of the carriage, and difference in heading were used together and interchangeably in this phase, based on logical assumptions. As a result of pre-evaluation, a MLPNNB using all these datasets was found to be the best scenario. Based on this outcome, the data was split into a training (70%) and a testing (30%) subset, then the MLPNNB was used to learn and generalize on these subsets. The main results indicate that the MLPNNB had an excellent performance, with a classification accuracy of*

[1] Department of Forest Engineering, Forest Management Planning and Terrestrial Measurements, Faculty of Silviculture and Forest Engineering, Transilvania University of Brasov, Şirul Beethoven 1, 500123, Brasov, Romania;
[2] National Forest Administration - RNP Romsilva, Maramureş Forest Directorate, 22 Decembrie Street 36, 430314, Baia Mare, Romania;
[3] Department of AGRARIA, Mediterranean University of Reggio Calabria, Feo di Vito snc, 89122, Reggio Calabria, Italy;
Correspondence: Stelian A. Borz; email: stelian.borz@unitbv.ro.

98.7, 98.4, and 98.8% in the pre-evaluation, training, and testing phases, respectively. Log-loss errors were also found to be very low (5, 5.9 and 4.1%, respectively), indicating a high generalization capability of the MLPNNB model. Based on the results, the main conclusion of the study is that original and derived GNSS data coupled with machine learning techniques could prove to be an important tool for operational monitoring and cable yarding efficiency model development, mainly due to the possibility of working with large amounts of data.

Key words: *steep terrain, time study, efficiency, automation, elemental, machine learning, classification, artificial intelligence, big data, Forestry 4.0*

1. Introduction

Cable yarding technology has been the backbone of steep terrain harvesting in many countries around the world [1], [18], [33], [36] and its use has been documented by a well-developed body of science. For instance, a review paper of Cavalli [4] has pointed out that a lot of effort has been put into evaluating the efficiency of cable yarding technology, environmental impact of operations, and simulation of cable systems. Efficiency assessment remains an important topic for cable-based systems, mainly because the developed models are used in research and operations management, a reason for which their reliability is of first importance [23]. However, factors such as the variability in operational conditions, practices and management of forests as specific to different regions may affect the shape of cable-based efficiency and productivity models, a reason for which additional model-developing work may be required either to improve the existing models or to develop models for new machines or operational setups.

Accurate models, on the other hand, may come at the expense of many resources when developed by traditional studies. This is because they are built from empirical data which needs to be collected in the right amount based on statistical grounds, in challenging environments by trained researchers [26]. These reasons have guided several scholars in researching new solutions to the problem of cable yarding efficiency assessment. Such solutions aimed at overcoming some of the mentioned limitations by automating the process of data collection, processing and analysis, at least in part, and they opened new doors for big data analysis. For instance, Gallo et al. [12] have developed an inference system to map and recognize the cycle time in cable yarding operations based on GNSS parameters such as movement direction and speed, by referencing their system to an external, arbitrarily-chosen point. They found high agreement between manually-recorded and system-output gross cycle time. Cheţa and Borz [6] have used a thresholding procedure to extract useful information for cable yarding from GPS signals and sound pressure level data. They were able to detect operational events such as moving and stopping of a carriage within an error or ± 2%. Since the productivity models require documentation of production (i.e., payload data), Guerra et al. [17] attempted to estimate it from GNSS data based on the measured deflection of a single-span cable yarder. More recently, Gallo et al. [13] have tested a GNSS-based system which was able to recognize 98% of the work cycles, with a difference in gross cycle time of 1% and the overall accuracy of separating work elements of less than 3%.

As all of the mentioned studies have used GPS-GNSS data, it is becoming obvious that this kind of data could assist in future new developments and in the improvement of the systems designed to recognize and classify operational events in cable yarding. This is mainly because GNSS data holds capabilities of documenting motorial (i.e., speed, acceleration, heading, distance etc.) and geometric (straightness, sinuosity etc.) descriptors which along with machine learning (ML) algorithms may turn into useful tools for the problem of transportation mode detection [39]. On the other hand, the European cable yarding systems were described to fulfill several technical functions, namely transportation, hoisting, skidding, handling, and energy transfer [18]. Among these, transportation functions such as moving the empty carriage from the landing to the stand, stopping and moving the loaded carriage from the stand to the landing, are essential, both for wood extraction and recognition of operations in the gross time domain. Compared to the GPS trajectories specific to other transportation modes, in a given setup cable yarding may restrict the variability of some motorial and geometric descriptors by providing a close to straight line between the tail spars, therefore a limited range in headings that could be collected by GPS, which ultimately may be seen as an important simplification in operational recognition. In turn, this simplification could enable the calculation of differential descriptors to make the input data invariant to a given geometric setup of the cable yarders, therefore would enable ML models built on a given cable yarding setup to work in recognizing operations as specific to other setups. The goal of this study was to evaluate if the GNSS-collected data could be useful in recognizing operational tasks in cable yarding by a machine learning approach, namely a Multi-Layer Perceptron Neural Network with Back propagation (MLPNNB). The study was based on the original and differentially-computed descriptors, and it was organized by considering the following objectives: *i)* checking which would be the best architecture of the MLPNNB for activity recognition and event classification in the time domain and *ii)* checking the capability of the selected MLPNNB to learn and generalize over the data in the time domain.

2. Materials and Methods

2.1. Data

The data used in this study was collected in 2017 by a Garmin® 60 stc GNSS unit which was placed on the carriage of a Wyssen sledge yarder that operated near the city of Zărneşti, Braşov County (Figure 1). The yarder was rigged to extract the wood downhill, a function that was assisted by gravity, while the sledge carrying the engine was rigged near the tail tree located in the uphill end of the corridor. Although the share of using cable yarding in Romania is low [27], due to its capability of saving fuel, this configuration of the yarding system seems to be dominant in low-access steep-terrain forests, as documented also in recent studies [28, 29]. The GNSS unit was set to sample locations at a rate of 0.2 Hz. Figure 1 shows the location of the yarder at the date of the field data collection, along with the relevant features of the cable yarder configuration.

The collected data covered one operational day, namely 8200 samples collected at a rate of five seconds (11.4 hours). This included the time needed to setup, place, and take down the GNSS unit from the carriage. Based on the GNSS data, the elevation of the carriage ranged between 867 (over the forest road at the unloading area, Figure 1b, group of locations 2) and 1,054 m (close to the uphill tail tree, Figure 1b, group of locations 4). The corridor was oriented at ca. 45° NE, and its length and average slope were of approximately 400 m and 47%, respectively. Most of the cable work during the observed day occurred in the first half of the corridor, although some turns were done by extracting wood from close to the uphill tail tree.

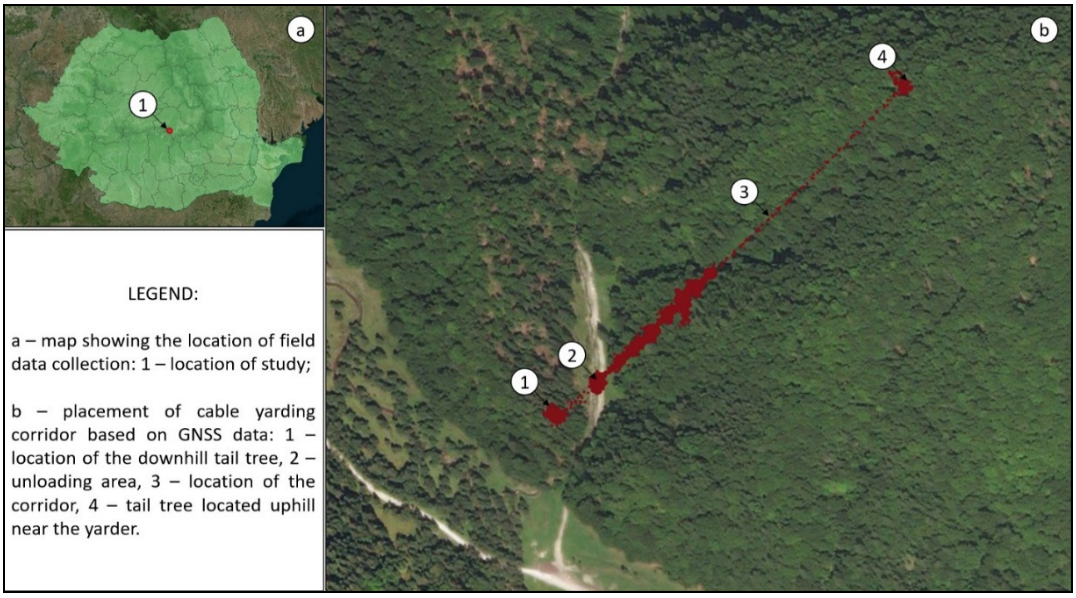

Fig. 1. *Location of the field study.*

Source: *maps built in QGis based on field collected GNSS locations and Bing® aerial data*

2.2. Data Processing

The collected data was imported in the form of a .gpx file into Garmin® BaseCamp® software from where the relevant features such as the observations' ID, elevation (hereinafter *e*, m), leg distance (hereinafter *d*, m), leg time (s), leg speed (hereinafter *s*, km/h), leg course (i.e., heading, hereinafter *c*, °), date and time, and position in coordinates, respectively, were copied into a Microsoft Excel® spreadsheet. As most of these parameters are output by BaseCamp® in the form of text data (i.e., along with their measurement units, in concatenated strings), Microsoft Excel® functions were used to parse the input data and to extract the relevant numeric information. The procedure was used for the *e*, *d*, *s*, and *c* variables. Once the database was available in this form, the non-relevant data covering the GNSS unit's placement and taking down was removed from the database. The removed data corresponded mainly to those data points located in the downhill direction from group 2 (Figure 1b) because the datapoints from group 1 (Figure 1b) were those located near the downhill tail tree, where the unit was placed and taken down from the carriage. Following the removal of this data, the resulted dataset contained 7821 observations, covering 39,105 seconds (ca. 10.9 hours). Once prepared, the data was subjected to some transformations, with the aim of making it invariant to the elevation and course characteristics as specific to the study location. Nevertheless, the following procedures were aimed also at preserving the variation brought about by changes in elevation due to the uphill and downhill movement of the carriage, and by the turning points at which the carriage engaged itself in the uphill or downhill movement. For this purpose, the difference in elevation was computed according to Equation 1, while the difference in course was computed according to Equation 2 for each observation in the dataset.

$$\Delta_{ei} = e_i - e_{i-1} \tag{1}$$

$$\Delta_{ci} = \left| c_i - c_{i-1} \right| \tag{2}$$

where: *e* and *c* stand for the elevation (m) and course (°), respectively, and *i* stands for a given (current) observation.

Three classes were differentiated in this study in relation to the carriage state, namely moving uphill (hereinafter MU), moving downhill (hereinafter MD), and carriage stopped (hereinafter S), as these three types of states commonly provide the information needed to characterize the gross work cycle time. All the observations were labeled to include them in ground truth classes, based on GNNS data imported in BaseCamp®. The software enables seeing a given selected location on the map as well as scrolling in the data point-by-point either manually or by playing the movement at a given speed. These actions update the location of a given data point and they allowed for a differentiation in the movement states of the carriage (MU, MD) as well as in detecting when the carriage was stopped either along the corridor or over the landing area. Although movement labelling was based on seeing evident changes in the distance followed by the carriage along a direction (course), there were few cases in which movement occurred successively in the uphill and downhill direction on relatively short distances, which could have been confused with potential errors caused by GNSS signal loss. Since these movements were oriented along the general heading of the corridor and they may occur in practice, the corresponding data was labelled as movement.

2.3. Data Analysis

2.3.1. Description of Data

A first step in the statistical analysis was the description of data, which was done by means of boxplot charts to characterize the input parameter data at the true class level. Accordingly, boxplots were developed for the *d*, Δ*e*, *s*, and Δ*c* variables. To characterize the class balance, observations from the true classes were described in terms of relative frequencies while the quality of the data was also characterized in the form of Signal to Noise Ratio (SNR) and the coefficient of variation (CV), by using the formulae described by Smith [35].

2.3.2. Data Pre-evaluation

Finding the best configuration of an ML model in terms of architecture and hyperparameters is a challenging task, which depends on the data structure. For MLPNNB models, the parameters that can be tuned are those related to the architecture, such as the number of hidden layers and hidden neurons per layer, and to the way of controlling the learning and generalization process, namely the activation function and solver, the regularization term (L2 penalty norm), and the number of iterations. When dealing with multi-modal signals or with the possibility of fusing data [5], supplementary questions may arise in what regards the potential contribution of the information carried by respective signals to the performance of the ML algorithm.

To infer the best configuration of the MLPNNB, all the available data (7821 observations) was used to train the algorithm by keeping the size of the network at its maximal values for the depth (number of hidden layers set at 3) and width (number of neurons per hidden layer set at 100), as enabled by the software used (see section 2.3.5). This was based on the findings and recommendations in previous studies [7], [16]. Irrespective of the input data used, the number of iterations was set at 1,000,000, and the model training was done by cross-validation based on a number of five folds. Other fixed parameters were the activation function used – ReLu (rectified linear unit) – which was chosen based on the recommendations of Goodfellow et al. [16] and the results reported in the relevant literature [25], [31], and the solver – ADAM (the stochastic gradient descent-based optimizer) which is one of the most recently-developed optimization algorithms standing out for its low training costs [22].

The parameter of the regularization term (α) is commonly used to avoid overfitting; increased values of α may help fixing high variance while decreased values may help fixing high bias [16], [37]. To tune the MLPNNB algorithm, values of 0.0001, 0.001, 0.01, 0.1, 1, 10, and 100 were used for α, which aimed, on the one hand, at fixing high bias and, on the other hand, at producing local minima and maxima in the performance metrics (see section 2.3.3) to help choose the best configuration of the MLPNNB for a given input dataset.

Several data input scenarios were considered to infer the best architecture of the MLPNNBB in which the datasets were chosen based on logical reasons. In a first scenario (hereinafter *S1*), variables d, Δe, s, and Δc were used as inputs in the MLPNNB and training of the algorithm was done by considering the varying values of α, as mentioned above. Then, the second (hereinafter *S2*, dataset: Δe, s, and Δc), third (hereinafter *S3*, dataset: Δe and s), fourth (hereinafter *S4*, dataset: s), and fifth (hereinafter *S5*, dataset: Δc) scenarios were designed to alter the number of input variables and to train the algorithm based on the same variation in the values of α. However, the choice of variables in these scenarios was based on their supposed capability of differentiating between the data contained in true classes.

As a final step in the pre-evaluation, the best input data scenario was selected by comparing the values of performance metrics (see section 2.3.3) returned at the overall level by the five tested scenarios. This was based on developing and analyzing graphs of the performance metrics returned by the tested data input scenarios against the regularization term.

2.3.3. Performance Metrics

The performance metrics used in this study were Area Under the Receiver Operating Characteristic Curve (hereinafter AUC), Classification Accuracy (hereinafter CA), F1 score (hereinafter F1), Precision (hereinafter PREC), Recall (hereinafter REC), Cross-Entropy (LogLoss, hereinafter LOGLOSS), and specificity (hereinafter SPEC). The definitions, properties, and interpretation of AUC, CA, F1, PREC, REC, and SPEC were discussed, for instance, in the papers of Fawcett [11] and Kamilaris and Prenafeta-Boldú [19]. The LOGLOSS metric is defined according to the equation from https://scikit-learn.org/stable/modules/generated/sklearn.metrics.log_loss.html [24]. These performance metrics were used both in the data pre-evaluation phase, where the aim was to identify the best configuration of the MLPNNB, and in the training and testing phases, where the aim was twofold, namely to see if the trained dataset has preserved the performance of that from the data pre-evaluation phase, as well as to check the generalization capability of the trained model. For these purposes, the maximum values of AUC, CA, F1, PREC, REC, and SPEC and the minimum value of the LOGLOSS performance metrics were criteria used to select the best scenario form sets S1-S5, as well as to examine the performance of the MLPNNB in the training and testing phase.

2.3.4. Training and Testing

For training and testing purposes, the dataset was split in two subsets of which one containing 70% of the data was used for training (hereinafter TRAIN) and the second, containing 30% of the data was used for testing (hereinafter TEST). Figure 2 shows the two partitions of the original dataset along with the three true classes plotted in the time domain. As shown, the events were relatively balanced in terms of frequency in the two datasets. Their occurrence in the time domain was also specific to the typical structure of a cable yarding work cycle, namely an empty turn (MU), stopping for lateral yarding (S), loaded turn of the carriage to the landing (MD), and stopping for lowering the load (S).

The best model architecture was then used to learn from the TRAIN dataset. Following model training and saving, it was tested on the TEST dataset. The performance metrics described in section 2.3.3 were used to evaluate the performance of the model following the training and testing phases.

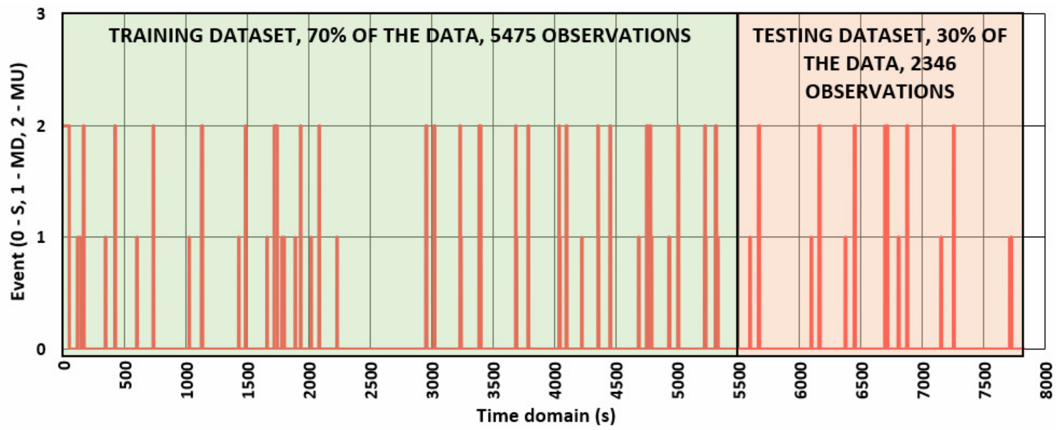

Fig. 2. *Partition of the original data into subsets for training and testing.*
Legend: *S – carriage stopped, MD – carriage moving downhill, MU – carriage moving uphill*

2.3.5. Computer Architecture and Software Used

The tasks of pre-evaluating the data as well as the tasks of training and testing the MLPNNB were run on a computer architecture that included the following features: system type - Alienware 17 R3, processor - Intel® Core™ i7-6700HQ CPU, 2.60GHz, 2592 MHz, 4 cores, 8 Logical Processors, installed physical memory (RAM) - 16 GB, operating system - Microsoft Windows 10 Home. BaseCamp® was the first software product used to import the GNSS data in the form of a .gpx file. The same application was used to move the data to Microsoft Excel® and to label the observations collected in the field. In addition to storing the data, Microsoft Excel® on which the Real Statistics package was installed was used to produce most of the artwork needed, as well as to sort the data and to apply functions for data parsing and transformation. The software used for data pre-evaluation, training, and testing in the form of MLPNNB was the Orange Visual Programming Software, 3.31.1 version [10], which holds the necessary functionalities for implementing Multi-Layer Perceptron models with Back Propagation, including data normalization if not specified otherwise by a given data processor. The software enables the creation of workflows designed for specific statistical tasks by means of interconnected widgets. In this study, Data, Neural Network, Test and Score, Save Model, Load Model, and Predictions widgets were used to pre-evaluate data, train and test the MLPNNB.

3. Results and Discussion

3.1. Description of Data

The total number of observations taken into the study accounted for 7821. Of these, 182 (2.3%), 281 (3.6%), and 7358 (94.1%) were labelled as MD (moving downhill), MU (moving uphill), and S (stopped), respectively. Looking at the data, it was found that the distribution of observations on true classes was highly unbalanced, with relatively close shares of carriage-moving classes (MD, MU) and a very high share of the stopped state (S). These results may be explained by two factors, namely the distances at which the carriage was moved, which were relatively short, as well as the speed at which the carriage moved during the empty and loaded turns. These factors commonly lead to shorter durations of carriage-moving work elements as compared to other tasks such as lateral skidding and attaching the load, therefore highly unbalanced classes are expected to also occur for increased extraction distances. For instance, Munteanu et al. [28] found shares of 7 and 9% of the cable yarding cycle time for uphill (empty) and downhill (loaded) movement of the carriage, for a similar configuration of the cable yarder which operated in thinning operations on an average extraction distance of ca. 300 m.

Figure 3 shows the main descriptive statistics of the parameters used as input data for classification. Leg distance (*d*, Figure 3a) averaged ca. 16, 11 and 1 m for moving downhill (MD), moving uphill (MU), and stopped (S) classes, results which are consistent with the practice of cable yarding operations as well as with the results shown for the moving speed (Figure 3c).

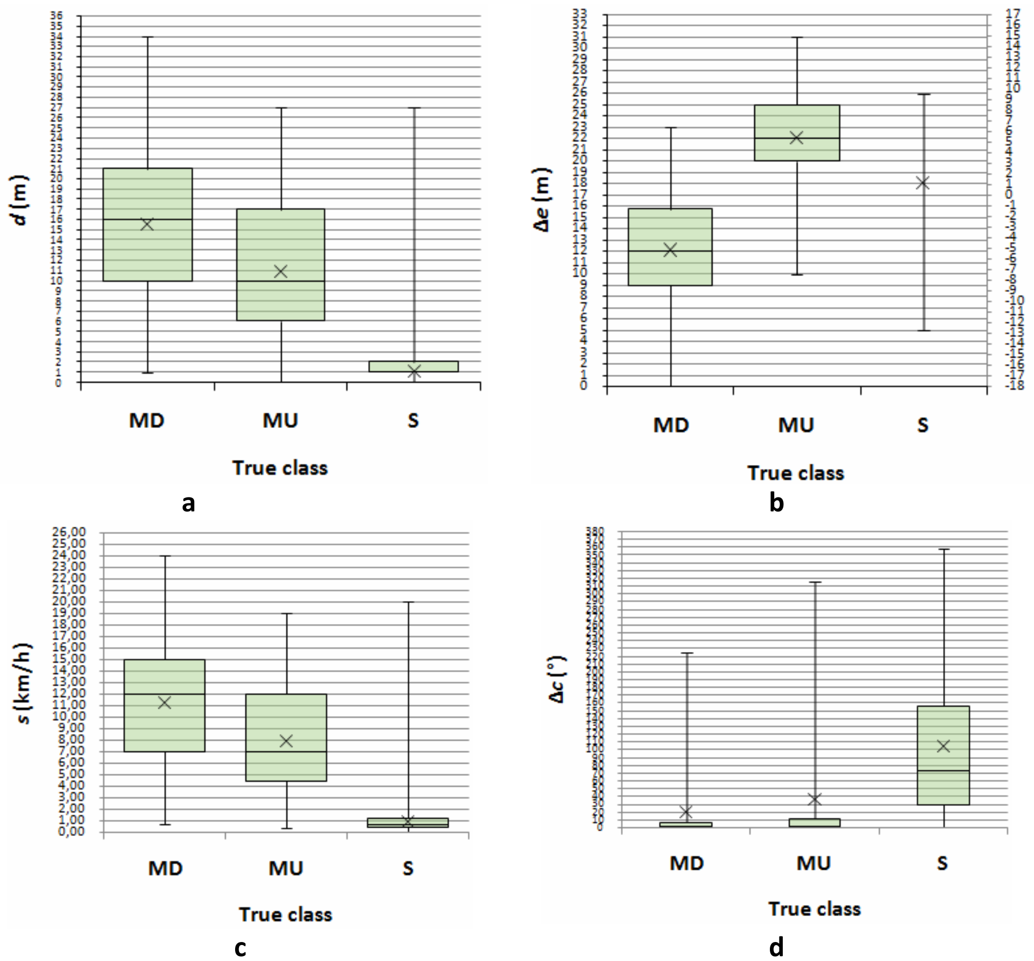

Fig. 3. *Class-level descriptive statistics of the parameters used as input data*

Concerning the leg distance and speed parameters, one could conclude that the average speed of the downhill movement was cca. 3 m/s (ca. 11 km/h, Figure 3c), while that of the uphill movement wasca. 2 m/s (8 km/h, Figure 3c), results which are similar to those reported in other studies characterizing the movement speed in loaded and empty turns [12]. In other studies which approached a cable yarding configuration similar to the one in this study, movement speeds were reported to be slightly higher [28, 29]. In what regards the Δ*c* parameter, it seems that the approach of calculating the differences in course (heading) was useful in delimitating carriage movement from stopped states more clearly. In this regard, some high values were preserved in the datasets of the MD and MU classes; however, the median and mean values were low as opposed to S. To summarize, based on the results shown in Figure 3, there were differences in the variability and central tendency of the data characterizing the true classes, facts which could concur with a potentially high classification performance. Nevertheless, there was also an evident interclass similarity, which is seen as a limitation for highly-accurate classification [3].

Table 1 summarizes the main statistics characterizing the quality of the signals used as data inputs for classification. Based on the results, the signals used as data input were rather noisy, with SNR values ranging from 0.42 to 2.06. Accordingly, the coefficients of variation (CV) had high values indicating a high intraclass variability.

Table 1

Signal to Noise Ratio (SNR) and the Coefficient of Variation (CV) of the parameters

Parameter	**(*d*, m)**			**(Δ*e*, m)***			***s* (km/h)**			**(Δ*c*, °)**		
True class	**MD**	**MU**	**S**	**MD**	**MU**	**S**	**MD**	**MU**	**S**	**MD**	**MU**	**S**
SNR	2.06	1.72	1.18	1.55	1.42	0.57	2.05	1.73	1.26	0.42	0.50	1.12
CV (%)	48.66	58.06	84.86	64.51	70.64	176.71	48.87	57.86	79.31	235.51	201.89	89.27

Note: *negative values were changed to their positive equivalents.

Excepting the Δ*c* parameter for which the SNR values were the highest in the case of the S class, the MD and MU classes returned higher values in this regard as compared to S, potentially showing a better suitability for a more accurate classification. However, the values of the SNR statistic were low irrespective of the class and parameter.

3.2. Pre-evaluation

Based on the data input scenarios and the hyperparameter tuning strategy described in section 2.3.2, a number of 35 data pre-evaluation tests were carried out, which accounted for an effective computing time of 21,204.288 seconds (ca. 5.9 hours). Although all the performance metrics (see section 2.3.3) were computed at both overall and class level and graphically represented against the variation of the regularization term, due to the limited space, Figure 4 shows only the variation of the most important ones at the overall level, namely the classification accuracy (CA), recall (REC), and log loss (LOG LOSS). Nevertheless, the following discussion is based also on the value of the rest of the performance metrics.

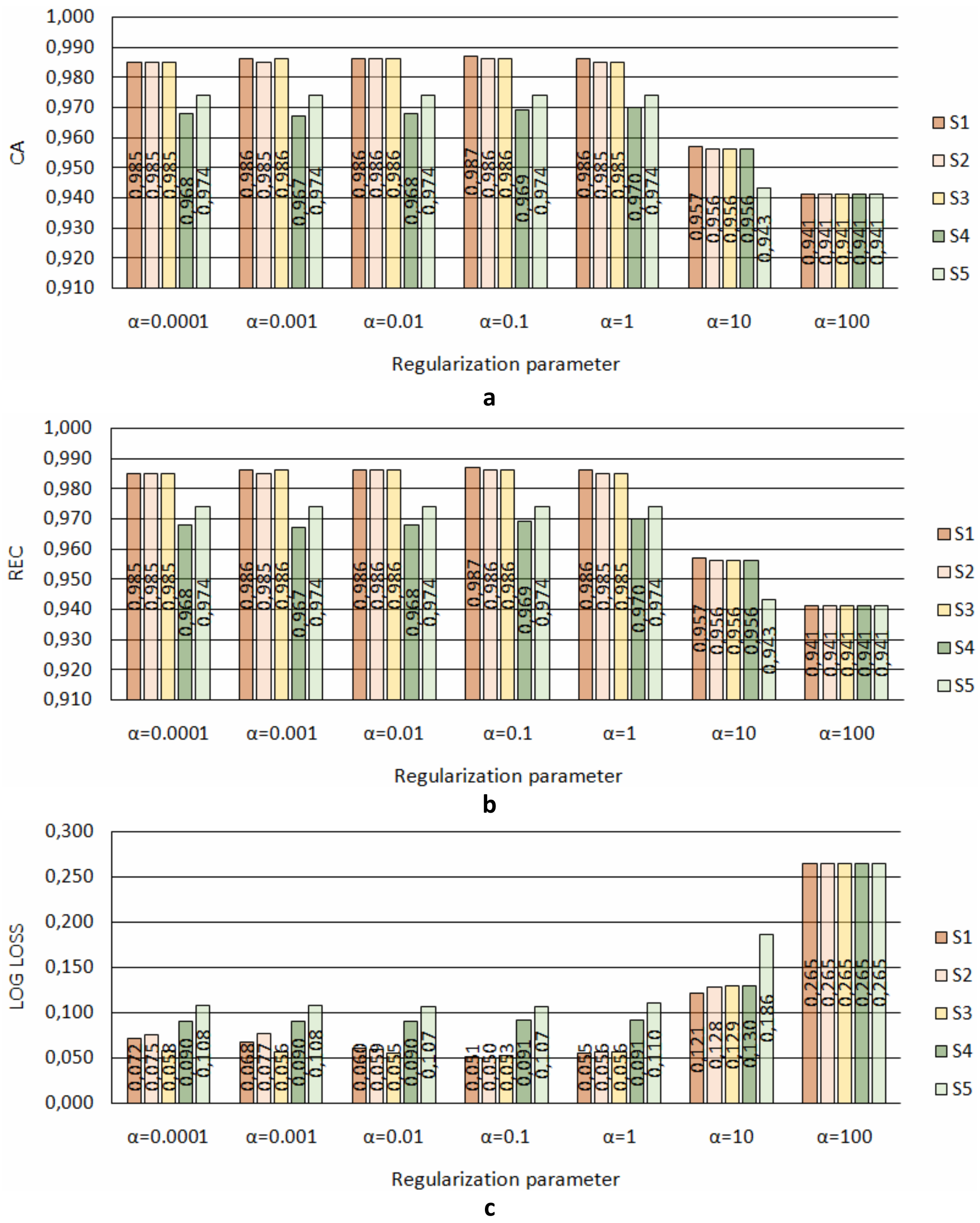

Fig. 4. *Classification accuracy, recall, and cross-entropy parameters as a function of the regularization term (α) and data input scenario (S1-S5)*

Looking at the data, the most accurate strategy of building the MLPNNB was that specific to scenario S1 which used as input data all the parameters (d, Δe, s and Δc) and a regularization term set at α=0.1. To describe the results, the highest values of the Area Under the Receiver Operating Characteristic Curve (AUC) were those returned by S1 and α set at 1 (AUC = 0.983) and 0.1 (AUC = 0.982), respectively (data not shown herein). Having in mind a minimal input of datasets as a criterion of selection when ties in values occurred, classification accuracy (CA) returned the highest overall values for S1 and α set at 0.1 (CA = 0.987) and 1 (CA = 0.986), respectively (Figure 4a). F1 metric had the highest values for S1 (α=0.1, 0.986) and S2 (α=0.1, 0.986), precision (PREC) for S1 (α=0.1, 0.986) and S2 (α=0.1, 0.986), and specificity for S1 (α=0.1, 0.872) and S2 (α=0.1, 0.860). Recall (REC, Figure 4b) returned the highest values for S1 (α=0.1, 0.987) and S2 (α=0.1, 0.986), while cross-entropy (LOG LOSS) returned the lowest values for S2 (α=0.1, 0.050) and S1 (α=0.1, 0.051).

Any of these performance metrics can be expressed as percents if multiplied by 100, therefore, in the pre-evaluation phase, the highest classification accuracy was of 98.7% (S1, α=0.1), the highest recall was of 98.7% (S1, α=0.1), and the lowest log loss was of 5% (S2, α=0.1). Accordingly, S1 with α set at 0.1 was selected as a strategy for training and testing the MLPNNB. For this scenario, Table 2 shows the values of performance metrics at overall and true class levels.

Table 2

Performance metrics of S1 (α = 0.1) in the pre-evaluation phase

Dataset and true classes	Performance metrics						
	AUC	**CA**	**F1**	**PREC**	**REC**	**LOGLOSS**	**SPEC**
Overall	0.982	0.987	0.986	0.986	0.987	0.051	0.872
MD	0.993	0.995	0.876	0.921	0.835	0.018	0.998
MU	0.977	0.989	0.845	0.887	0.808	0.038	0.996
S	0.985	0.989	0.994	0.991	0.997	0.041	0.864

As shown, the classification accuracy, which is the ratio of correctly classified instances as true positives and negatives to all the instances of a dataset (true positives, true negatives, false positives, and false negatives), was close to 100% in the case of the MD class and it was close to 99% in the case of the MU and S classes. However, the recall parameter, which is the ratio of positives correctly classified to all the positives, was of ca 88, 85, and 99% in the case of MD, MU, and S, respectively. The lowest log-loss error was that of the MD (0.018) class while it was relatively the same for the MU (0.038) and S (0.041) classes.

3.3. Training and Testing of the MLPNNB

Excepting the log loss, following the training phase (Figure 5), the values of all the performance metrics decreased as compared to those returned by the pre-evaluation tests. As such, a performance dilution occurred due to changing the quantity of underlying data. However, the changes were minor, accounting for 0.3% in the case of CA and REC. Log loss error increased in the training dataset by 0.8%.

Altogether, these small differences supported the attempt to pre-evaluate the data with the aim of building performant MLPNNB models for training and testing purposes. This was also emphasized by the results on performance metrics obtained in the testing phase (Figure 5), which showed values higher by 0.1 to 0.2% (CA, REC) and lower by 1% (LOGLOSS). Therefore, the classification performance increased as the generalization error decreased in the testing dataset.

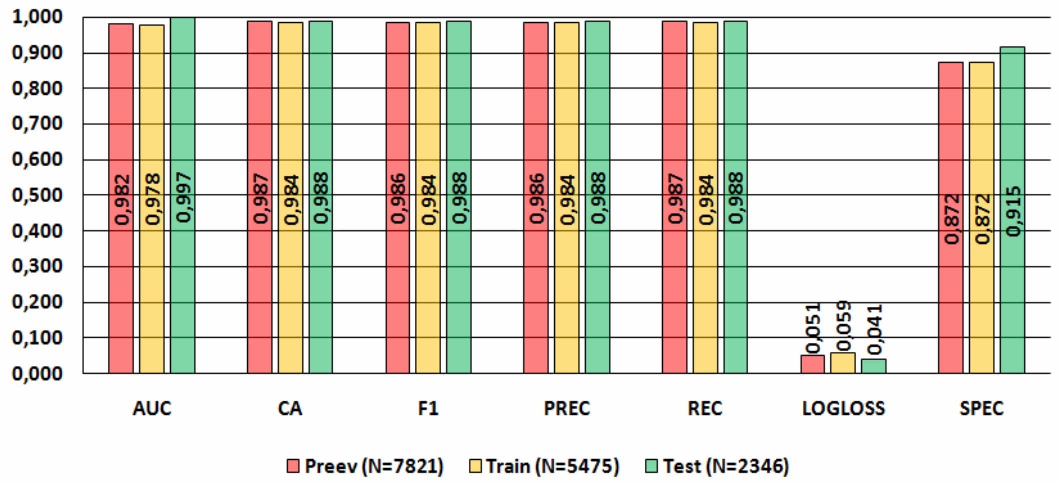

Fig. 5.*Comparison of the pre-evaluation, training, and testing performance metrics (S1, α=0.1)*

To prove that the performance of classification actually depended on the inclusion of all parameters, Figure 6 was prepared to show the probabilities associated with correctly classifying given instances into the three classes. For example, high values of *d* (up to 30 m, Figure 6a) indicate lower probabilities of classifying a given instance as S and higher probabilities of classifying it as MD or MU. Higher differences in elevation (Figure 6b) were more likely to classify an instance as moving.

Same held true for speed, as there were lower probabilities to classify an instance as S for high speeds and vice versa, to classify instances as MD and MU when the speed was low (Figure 5c). In what regards the difference in heading (Figure 6c), the probability of classifying an instance as S increased as the value of Δ*c* increased and vice versa, as the value of Δ*c* decreased, the probability of classifying a given instance as movement generally decreased. Altogether, the data shown in Figure 6 proves the utility of the selected parameters in training and testing accurate MLPNNB models to differentiate between given work elements. However, Figure 6 also proves the fact that a high classification performance may be achieved only by using these parameters in conjunction.

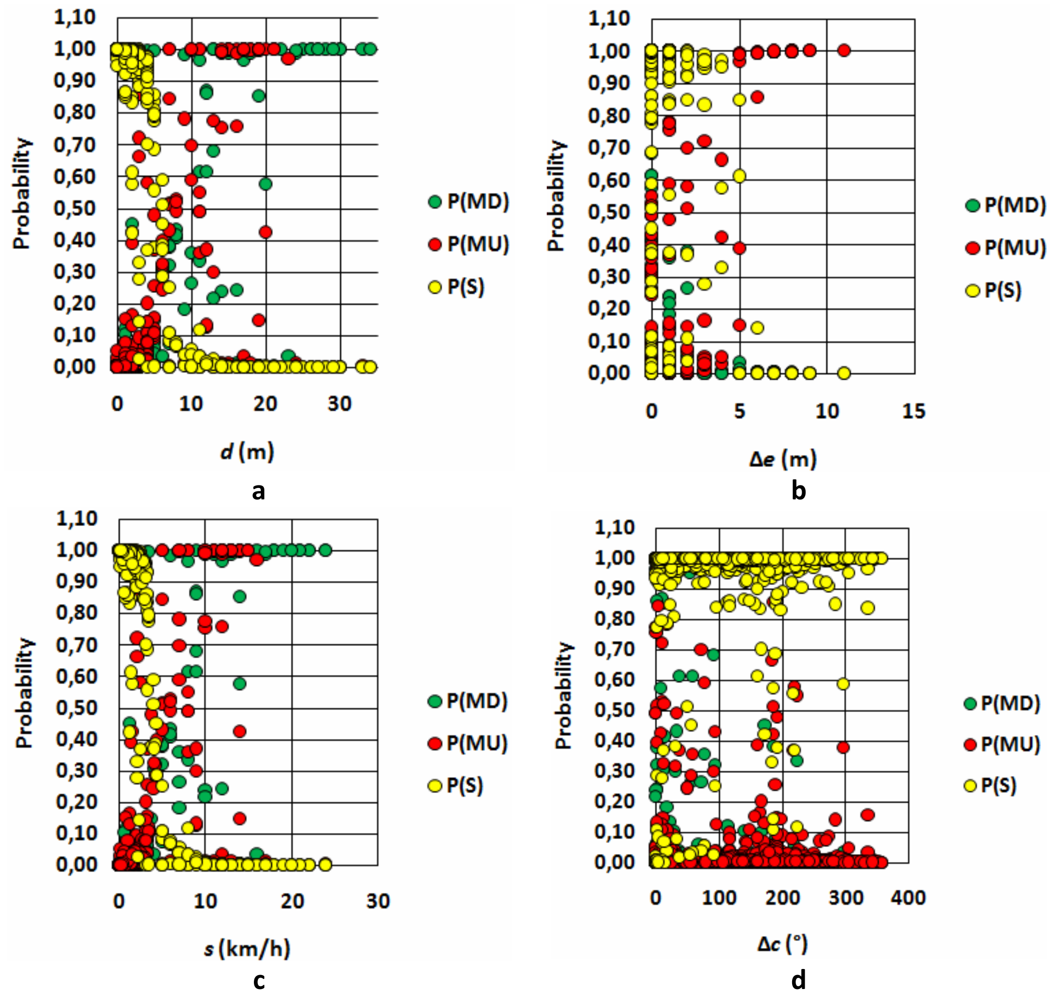

Fig. 6.*Probability (P) of correctly classifying given instances in the MD (moving downhill), MU (moving uphill), and S (stopped) classes plotted against the variation of leg distance (d, m), difference in elevation (Δe, m), speed (s, km/h), and difference in course (Δc, °), as returned by the testing phase*

3.4. Discussion

While there is a high body of literature on cable yarding performance assessment by various techniques, the discussion given in the following aims at synthetizing the results of this study in the view of similar approaches to the problem. In what regards the features used to feed the MLPNNB, the study has found that all of them were important contributors to the classification performance. Based on GNSS data, several studies reported a range of carriage moving speeds, starting from ca. 2 and ending with ca. 5m/s, and all of them agreed that two classes of speed could be differentiated as empty and loaded moving of the carriage [12, 13], [28, 29]. Accordingly, the speed of the carriage would remain an important parameter in developing automated event recognition systems in cable yarding, mainly because it carries the information needed to distinguish between the stopped and moving states and between the two main types of carriage movements. In this study, the speed parameter played an important role in classification accuracy and it was interrelated to the leg distance parameter with which it was highly correlated (data not shown herein). On the other hand, the contribution of each parameter taken individually, or in association as smaller subsets built on logical reasons has been proved to be a less efficient strategy to increase the classification accuracy and generalization ability of the MLPNNB in the pre-evaluation phase. Hence, the combined use of all the parameters could provide increased classification accuracies as compared to what could be expected by using them alone or in combinations that exclude some of them.

This study is based on a specific configuration of the cable yarder which was rigged to extract the wood downhill, a function which was assisted by gravity. There are many other rigging configurations in which a cable yarder may work, including those used to extract the wood uphill or to work on flat terrain [14, 15], [32], [38]. These commonly used configurations need to be accounted for in the features used to develop an operational recognition system with the aim of making the statistical learning algorithm invariant to given setups. While the speed and leg distance were kept in the form provided by the GNSS data, elevation and course were transformed by differentiation in this study. Then, assuming the existence of a good GNSS signal, the two moving states could be learned by differentiating the elevation of a given location in relation to the previous one, while using the absolute value of elevation would have probably led to learning based on a given cable yarder configuration, therefore to a limited capability in generalizing on new datasets. As a consequence, for uphill yarding there will be changes in the sign of differentiated values while in flat terrains the differences would probably be close to zero. Since this study has used the signed differences in elevation, future studies could improve the approach by removing this issue. Most probably this could be solved by using an equation in the form of Equation 2 (see section 2.2), which was used to detect in the course data those segments which were characterized by relatively straight trajectories and which are typical for movement. In addition, Equation 2 generalizes these types of trajectories by removing the geographical heading context. In essence, values close to zero according to Equation 2 may indicate movement. Since itis difficult to have perfectly flat terrains or error-free GNSS signals, the contribution of speed to the learning and generalizing capabilities of the MLPNNB models could prove to be a factor of first importance. Assuming, for instance, a perfectly straight line followed by the GNSS unit during movement and no GNSS location errors, in theory the difference in elevation would be null. This means that MLPNNB would have to rely heavily on speed to detect movement. Using only the speed in this study (pre-evaluation) returned a high classification performance in general, but the value of recall metric was lower for movement events (data not shown herein).

Data labelling effort and GNSS sampling rates could also be improved by running comparative studies. Video surveillance, for instance, could be useful to document the events in more detail, assuming that a video camera could be placed at a safe location on the carriage. Although processing effort to label the GNSS data based on video files could be high [9], [30], it will provide more detailed knowledge which could be helpful for machine learning models in understanding the patterns in data. Further studies could pursue this idea so as to provide better operational recognition models. In a previous study, other modalities were used to better understand and label the data [6] but their capability may be influenced by other underlying processes. In addition, the quantity of data used to train and test machine learning algorithms is of first importance. It needs to cover the variability brought by operational setups and conditions, therefore further studies could attempt to include these factors in more accurate models by extending data collection, processing, and analysis. In this regard, one important advantage of the machine learning algorithms is that once robustly trained, they can be used with minimal effort to classify the events on countless datasets.

The values of the performance metrics returned in the pre-evaluation, training, and testing phases of the MLPNNB could be seen as excellent compared to the general body of knowledge gained in applications of deep learning [19]. Although several problems of forest operations were approached successfully by machine learning [2], [8], [21], [34], no similar studies were found to compare the classification performance metrics returned by this study. Nevertheless, the AUC was found to be higher than 0.97 in all three phases. According to Fawcett [11] this parameter is important in evaluating the performance of classifiers based on receiver operating characteristics (ROC) graphs, and it maps the performance of a classifier in a bidimensional space as a function of true positive rate (sensitivity) and false positive rate (1 – specificity). Classifiers which approach a high sensitivity (close to 1) and a low false positive rate (close to 0) are seen to have the best performance. Values of AUC close to 1 indicate high performances of their corresponding classifiers. Classification accuracy (CA) returned values higher than 98%, irrespective of the evaluation phase. This metric maps the ratio of instances predicted as true positives and false negatives over all the predictions [19] which, in simple words, is the ratio of correct predictions to the total of predictions. In the testing phase, for instance, this ratio approached a value of 0.99. This means that only 1% of the data was misclassified in the testing phase (ca. 23 instances out of 2346 observations). Similar values were found for the recall metric, meaning that close to 99% of the positives were classified correctly. In addition, the log-loss errors were low and similar in value among the three evaluation phases, meaning that the MLPNNB algorithm trained and generalized well.

The width and the depth of the MLPNNB model were based in this study on recommendations formulated in the relevant literature and on findings of other studies. Although the maximal size as permitted by the software used has led to excellent results, it does not mean that it would be the optimal one, particularly in terms of training time. The same applies to the number of iterations, which was kept at maximum although it might not be the optimal one.

GNSS data has been proven to be particularly useful in forestry-related applications [20]. This includes applications to operational monitoring and task recognition as proved by this study and the previous ones. Although the model of this study was built just for proving a concept, the underlying methodology can be extended by including more data containing similar parameters as it becomes available, or to other models starting from newly documented data. In addition, such models could be integrated as an intelligent operational monitoring component in Geographic Information Systems (GIS) which are able to exploit and bring to the model other parameters such as those characterizing the topography.

4. Conclusion

The goal of this study was to evaluate whether the GNSS-collected data could provide useful information for recognizing operational tasks in cable yarding by a Multi-Layer Perceptron Neural Network with Backpropagation (MLPNNB). The results of the pre-evaluation, training, and testing phases of this study indicate that the approach is both possible and feasible under such circumstances in which a proper set of descriptors is used, of which some were computed by differentiation. Accordingly, the classification performance was found to be excellent in terms of accuracy, true positive rate (recall), and log-loss error, facts which indicate that the model was able to learn and generalize on GNSS-based data very well. The results also indicate that less-resource intensive approaches to the problem of developing efficiency models in cable yarding are becoming readily available. In turn, this will enhance our ability of extending the utility of existing models or building new ones adapted to new operational conditions. Future research should focus on documenting the GNSS data in more detail, probably by labelling it based on information collected as video files, and on getting it at finer sampling rates which will enable the models in gaining more knowledge from the data structure. To this end, the methodology used in this study to prove a concept could be extended and used to build more accurate models.

References

1. Bont L., Heinimann H.R., 2012. Optimum geometric layout of a single cable road. In: European Journal of Forest Research, vol. 131(5), pp. 1439-1448. DOI: 10.1007/s10342-012-0612-y.
2. Borz S.A., Păun M., 2020. Integrating offline object tracking, signal processing and artificial intelligence to classify relevant events in sawmilling operations. In: Forests, vol. 11(12), ID 1333. DOI: 10.3390/f11121333.
3. Bulling A., Blanke U., Schiele B., 2014. A tutorial of human activity recognition using body-worn inertial sensors. In: ACM Computing Surveys, vol. 46(3), ID 33. DOI: 10.1145/2499621.
4. Cavalli R., 2012. Prospects of research on cable logging in forest engineering community. In: Croatian Journal of Forest Engineering, vol. 33(2), pp. 339-356.
5. Chen K., Zhang D., Yao L. et al., 2018. Deep learning for sensor-based human activity recognition: Overview, challenges and opportunities. In: J.

ACM, vol. 37(4), ID 111. DOI: 10.1145/1122445.1122456.

6. Cheţa M., Borz S.A., 2017. Automating data extraction from GPS files and sound pressure level sensors with application in cable yarding time and motion studies. In: Bulletin of the Transilvania University of Brasov, Series II: Forestry, Wood Industry, Agricultural Food Engineering, vol. 10(59), no. 1, pp. 1-10.
7. Cheţa M., Marcu M.V., Borz S.A., 2020a. Effect of training parameters on the ability of artificial neural networks to learn: a simulation on accelerometer data for task recognition in motor-manual felling and processing. In: Bulletin of the Transilvania University of Brasov, Series II: Forestry, Wood Industry, Agricultural Food Engineering, vol. 13(62), no. 1, pp. 19-36. DOI: 10.31926/but.fwiafe.2020.13.62.1.2.
8. Cheţa M., Marcu M.V., Iordache E. et al., 2020b. Testing the capability of low-cost tools and artificial intelligence techniques to automatically detect operations done by a small-sized manually driven bandsaw. In: Forests, vol. 11(7), ID 739. DOI: 10.3390/f11070739.
9. Contreras M., Freitas R., Ribeiro L. et al., 2015. Multi-camera surveillance systems for time and motion studies of timber harvesting equipment. In: Computer and Electronics in Agriculture, vol. 135, pp. 208-215. DOI: 10.1016/j.compag.2017.02.005.
10. Demsar J., Curk T., Erjavec A. et al., 2013. Orange: Data Mining Toolbox in Python. In:Journal of Machine Learning Research, vol. 14, pp. 2349-2353.
11. Fawcett T., 2006. An introduction to ROC analysis. In: Pattern Recognition Letters, vol. 27, pp.861-874, DOI: 10.1016/j.patrec.2005.10.010.
12. Gallo R., Grigolato S., Cavalli R. et al., 2013. GNSS-based operational monitoring devices for forest logging operation chains. In: Journal of Agricultural Engineering, vol. XLIV(s2), pp. 140-144. DOI: 10.4081/ jae.2013.269.
13. Gallo R., Visser R., Mazzetto F., 2021. Developing an automated monitoring system for cable yarding systems. In: Croatian Journal of Forest Engineering, vol. 42(2), pp. 213-225. DOI: 10.5552/crojfe.2021.768.
14. Ghaffariyan M.R., Stampfer K., Sessions J., 2009. Production equations for tower yarders in Austria. In: International Journal of Forest Engineering, vol. 20(1), pp. 17-21. DOI: 10.1080/14942119.2009.10702571.
15. Ghaffaryian M.R., Stampfer K., Sessions J., 2010. Optimal road spacing of cable yarding using a tower yarder in Southern Austria. In: European Journal of Forest Research, vol. 129(3), pp. 409-416. DOI: 10.1007/s10342-009-0346-7.
16. Goodfellow J., Bengio Y., Courville A., 2016. Deep learning. MIT Press, 2016. Available at: https://www. deeplearningbook.org/. Accessed: 10th of March, 2022.
17. Guerra F., Marchi L., Grigolato S. et al., 2020. Indirect payload estimation in cable-logging operations. In: Engineering for Rural Development, Jelgava, 20-22.05.2020, pp. 1914-1919. DOI: 10.22616/ERDev.2020.19. TF537.
18. Heinimann H., Stampfer K., Loschek J. et al., 2001. Perspectives on Central

European cable yarding systems. In: International Mountain Logging and 11th Pacific Northwest Skyline Symposium, Seattle (WA), College of Forest Resources, University of Washington and IUFRO.
19. Kamilaris A., Prenafeta-Boldú F.X., 2018. Deep learning in agriculture: A survey. In: Computers and Electronics in Agriculture, vol. 147, pp. 70-90. DOI: https://doi.org/10.1016/j.compag.2018.02.016.
20. Keefe R.F., Wempe A.M., Becker R.M. et al., 2019a. Positioning methods and the use of location and activity data in forests. In: Forests, vol. 10(5), ID 458. DOI: 10.3390/f10050458.
21. Keefe R.F., Zimbelman E.G., Wempe A.M., 2019b. Use of smartphone sensors to quantify the productive cycle elements of hand fallers on industrial cable logging operations. In: International Journal of Forest Engineering, vol. 30(2), pp. 132-143. DOI: 10.1080/14942119.2019.1572489.
22. Kingma D.P., Ba J.L., 2015. ADAM: A method for stochastic optimization. In: 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015.
23. Lindroos O., Cavalli R., 2016. Cable yarding productivity models: a systematic review over the period 2000-2011. In: nternational Journal of Forest Engineering, vol. 27(2), pp. 79-94. DOI: 10.1080/14942119.2016.1198633.
24. Log loss, aka logistic loss or cross-entropy loss. Available at: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.log_loss.html. Accessed: 09th of March, 2022.
25. Maas A.L., Hannun A.Y., Ng A.Y., 2013. Rectifier nonlinearities improve neural network acoustic models. In: Proceedings of the 30th International Conference on Machine Learning, ICML 2013, Atlanta, GA, USA, June 16-21, 2013.
26. Magagnotti N., Spinelli R. (eds.), 2012. Good practice guidelines for biomass production studies. Sesto Fiorentino: COST Action FP-0902 and CNR IVALSA, 53 p.
27. Moskalik T., Borz S.A., Dvorak J. et al., 2017. Timber harvesting methods in Eastern European countries: a review. In: Croatian Journal of Forest Engineering, vol. 38(2), pp. 231-241.
28. Munteanu C., Ignea G., Akay A.E. et al., 2017. Yarding pre-bunched stems in thinning operations: estimates on time consumption. In: Bulletin of the Transilvania University of Brasov, Series II: Forestry, Wood Industry, Agricultural Food Engineering, vol. 10(59), no. 1, pp. 43-54.
29. Munteanu C., Yoshida M., Iordache E. et al., 2019. Performance and cost of downhill cable yarding operations in a group shelterwood system. In: Journal of Forest Research, vol. 24(3), pp. 125-130. DOI: 10.1080/13416979.2019.1603577.
30. Muşat E.C., Apăfăian A.I., Ignea G. et al., 2016. Time expenditure in computer aided time studies implemented for highly mechanized forest equipment. In: Annals of Forest Research, vol. 59(1), pp. 129-144. DOI: 10.15287/afr.2015.473.
31. Nair V., Hinton G.E., 2010. Rectified linear units improve restricted Boltzmann machines. In: Proceedings of the 27th International Conference

on Machine Learning (ICML 2010), Haifa, Israel, June 21-24, 2010.

32. Oprea I., 2008. Tehnologia exploatării lemnului (Timber technology [in Romanian]). Transilvania University Publishing Press, Brasov, 237 p.
33. Proto A.R., Skoupy A., Macrì G. et al., 2016. Time consumption and productivity of a medium size mobile tower yarder in downhill and uphill configurations: a case study in Czech Republic. In: Journal of Agricultural Engineering, vol. 47(4), pp. 216-221. DOI: 10.4081/jae.2016.551.
34. Proto A.R., Sperandio G., Costa C. et al., 2020. A three-step neural network artificial intelligence modeling approach for time, productivity and costs prediction: A case study in Italian forestry. In: Croatian Journal of Forest Engineering, vol. 41(1), pp. 35-47. DOI: 10.5552/crojfe.2020.611.
35. Smith S.W., 2003. Digital signal processing. A practical guide for engineers and scientists. Newness, imprint of Elsevier Science, 650 p.
36. Stoilov S., Proto A.R., Angelov G. et al., 2021. Evaluation of salvage logging productivity and costs in sensitive forests of Bulgaria. In: Forests, vol. 12(3), ID 309. DOI: 10.3390/f12030309.
37. Varying regularization in Multi-layer perceptrons. Available at: https://scikit-learn.org/stable/autoexamples/neural_networks/plot_mlp_alpha.html#sphx-glr-auto-examples-neural-networks-plot-mlp-alpha-py. Accessed: 09th of March, 2022.
38. Visser R., Harrill H., 2017. Cable yarding in North America and New Zealand: A review of developments and practices. In: Croatian Journal of Forest Engineering, vol. 38(2), pp. 209-217.
39. Yang X., Stewart K., Tang L. et al., 2018. A review of GPS trajectories classification based on transportation mode. In: Sensors, vol. 18, ID 3741. DOI: 10.3390/s18113741.