

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900

Open access books available

185,000

International authors and editors

200M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



AI-Based Approach for Lawn Length Estimation in Robotic Lawn Mowers

Yoichi Shiraishi, Haohao Zhang and Kazuhiro Motegi

Abstract

This chapter describes a part of autonomous driving of work vehicles. This type of autonomous driving consists of work sensing and mobility control. Particularly, this chapter focuses on autonomous work sensing and mobility control of a commercial electric robotic lawn mower, and proposes an AI-based approach for work vehicles such as a robotic lawn mower. These two functions, work sensing and mobility control, have a close correlation. In terms of efficiency, the traveling speed of a lawn mower, for example, should be reduced when the workload is high, and vice versa. At the same time, it is important to conserve the battery that is used for both work execution and mobility. Based on these requirements, this chapter is focused on developing an estimation system for estimating lawn grass lengths or ground conditions in a robotic lawn mower. To this end, two AI algorithms, namely, random forest (RF) and shallow neural network (SNN), are developed and evaluated on observation data obtained by a fusion of ten types of sensor data. The RF algorithm evaluated on data from the fusion of sensors achieved 92.3% correct estimation ratio in several experiments on real-world lawn grass areas, while the SNN achieved 95.0%. Furthermore, the accuracy of the SNN is 94.0% in experiments where sensor data are continuously obtained while the robotic lawn mower is operating. Presently, the proposed estimation system is being developed by integrating two motor control systems into a robotic lawn mower, one for lawn grass cutting and the other for the robot's mobility.

Keywords: AI, robotic lawn mower, work vehicle, Random Forests, Neural Network, embedded system, Hybrid Twin

1. Introduction

Recently, automated driving algorithms and systems for work vehicles such as robotic lawn grass or grass mowers (robo-mowers) [1–3], autonomous snow blowers [4], automatic guided vehicles (AGVs) [5], autonomous delivery vehicles [6], and autonomous mobile robots (AMRs) [7], have attracted much attention. These vehicles are made possible by significant advances in sensor fusion technology, high-performance embedded systems, AI algorithms and advanced model-based design or development methods. Particularly, Industry 4.0 or Society 5.0 needs the digital transformation or smart factory in industry and, now, AGVs and AMRs perform some tasks that are essential for constructing the automatic

production lines. As these vehicles share a battery for their work and mobility, the interaction between their functions should be effectively controlled to reduce battery charging frequency and time, as well as working time. The authors in [4] attempted to optimize the skidding control of a snow blower, which has a motor for mobility and an engine for blowing snow. In this case, the engine is also used to generate electric power, which is then used to charge the battery that powers the motor. In this sense, its work and mobility share the energy, thus the two functions require effective energy management. Precise control handling of work load in these work vehicles is critical for optimizing its energy management.

In the following sections, because commercial robo-mowers [8] are popular and readily available for experiments, a robo-mower is used as an example for optimizing energy management in work vehicles. Mobility control is the next research theme for optimizing the energy management of robo-mowers. Current robo-mowers do not recognize the length of lawn grasses or ground conditions such as dirt, gravel, or concrete. As a result, the motor for cutting lawn grasses operates at a constant rotation speed from start to finish. Therefore, if the rotation speed of the motor for a lawn grass cutter is precisely controlled, battery wastage can be avoided. Moreover, because the control of grass cutting and mobility is correlated, the mobility speed should be controlled according to the lawn grass lengths and ground conditions. Then, the working time can also be reduced. Therefore, the precise estimation of lawn grass lengths using effective sensor data is required in the first stage, i.e., preventing battery wastage. Then, in the second step, the mobility of robo-mowers is controlled according to the estimation results from the first stage. Finally, a cooperative control of a group of robo-mowers is researched [3] and implemented. In particular, the group control of robo-mowers becomes meaningful when the performance of each robo-mower is optimized.

In this study, an AI-based approach is adopted for the estimation of lawn grass lengths from the fusion of sensor data. A random forest (RF) algorithm and shallow neural network (SNN) are suggested. Ten measurement data types are obtained from sensors attached to a robo-mower. The combination of sensor data types is essential for lawn grass estimation, that is, a sensor fusion problem is discussed. In general, the sensor fusion and use of big data have attracted many researchers' interest. Recently, there have been detailed surveys on the combination of sensor fusion and big data analysis [9, 10]. Some applications to actual problems have also been reported [11–13]. The popular approach for big data analysis is the use of machine learning. Takami G., et al. [11] studied the observation of plant status. They used three kinds of sensors and a deep learning (DL) algorithm for big data analysis. The details of the DL are not described, and the processing time of the observation system is not known; however, it may be useful to learn that they predicted the deterioration of sensors performance through their combination. Alonso S., et al. [12] also adopted the same approach for observing a screw compressor in a chiller. They used five kinds of sensor data and a 1D convolutional neural network (CNN) for their analysis. The adoption of 1D CNN makes monitoring faster and real-time processing is realized. Their approach is probably suitable for data without any estimated features; however, in this study some features may be efficient for the estimation task in advance. Li C., et al. [13] performed the diagnosis of rotating machinery. They used vibration sensor signals, and the Gaussian-Bernoulli deep Boltzmann machine was used for their analysis. The accuracy of fault estimation was evaluated; however, its real-time processing requirement was not mentioned. Therefore, this approach cannot be applied to the problem dealt with in the following.

In the experiments of the proposed AI-based approach, the application of RF algorithm to the fusion of seven sensors attained a 92.3% correct estimation ratio in

several experiments on actual lawn grass areas. In addition, the application of SNN attained 95.0%. Moreover, the accuracy of SNN is 94.0% in such trials in which sensing data are continuously obtained while the robo-mower is in operation. The proposed estimation system is being developed by integrating it with two types of motor control systems for grass cutting and robot mobility, respectively. At present, the authors are promoting the research and development of robo-mowers for commercial use by collaborating with an automobile company.

The outline of this chapter is as follows. Section 2 describes the Hybrid Twin TM, which is the basic idea for controlling the robo-mower in real time. Section 3 describes robo-mower used in this chapter; however, the discussions are not limited to this robo-mower. Moreover, the estimation problem of lawn grass lengths is also defined in this section. Section 4 describes the proposed RF and SNN algorithms. Section 5 describes the experimental results based on the big data obtained from sensor fusion and a set of features for classifying the sensor data are. Furthermore, the set of necessary sensors and performance evaluations of the proposed algorithms are stated. Section 6 describes the evaluation of the proposed SNN algorithm when applied to the consecutive sensor data obtained in real-world use. Finally, the chapter is summarized in Section 7.

2. Hybrid twin within work vehicle

The Hybrid Twin TM approach [14] is efficient for real-time object control. The proposed estimation method is useful for controlling the operation of lawn grass cutter motor and a mobility motor. When the robot is operating in an area with long lawn grasses, the motor should be set to the maximum rotation speed. On the other hand, the rotation speed should be reduced or stopped when the robo-mower is operating in an area with short lawn grasses or in an area without any lawn grasses, respectively. As a result, battery consumption will decrease. Furthermore, if it is possible to control the robo-mower's speed so that it decreases or increases according to the length of lawn grasses, the working time will be greatly reduced. When a ground without lawn grasses is identified, laying the electric cable that defines the boundary of the area is no more necessary and, as a result, the required maintenance is reduced.

Digital Twin has become popular for implementing smart factory, and it has been used [15] for controlling mission-critical systems, such as nuclear plant, airplane control, or rocket control in the aerospace industry. The Digital Twin constructed in the virtual space means a twin of a real space object. The twin is a precise model, and its behaviors are reproduced in the virtual space. The Hybrid Twin TM is an extension of Digital Twin. As the target system has become large and complicated, the Virtual Twin has been separated from the Digital Twin, as shown in **Figure 1**.

The Digital Twin only obtains data from a fusion of sensors, and measurement data with some abstractions are transferred to the Virtual Twin. The Virtual Twin is a precise co-simulator consisting of subsystems obtained using a model-based design method. This simulator must be sufficiently fast, and it is usually a 1-D simulator, which is a high-speed version of the original 3-D simulator is used. The Hybrid is a combination of the Virtual Twin and Digital Twin, and the optimized state of the real system on time $(t + \Delta t)$ must be feedback to the real system from the state on time (t) . This loop is iterated over with the time interval (Δt) . As a result, the state of the real system is optimized in real time. Measurement data are extracted from the fusion of sensors for robo-mower operations, and noise reduction is applied to the obtained parameters in the Digital Twin. This means that the Digital Twin is an accurate numerical model of real objects. The Virtual Twin receives the obtained data $s(t)$ at

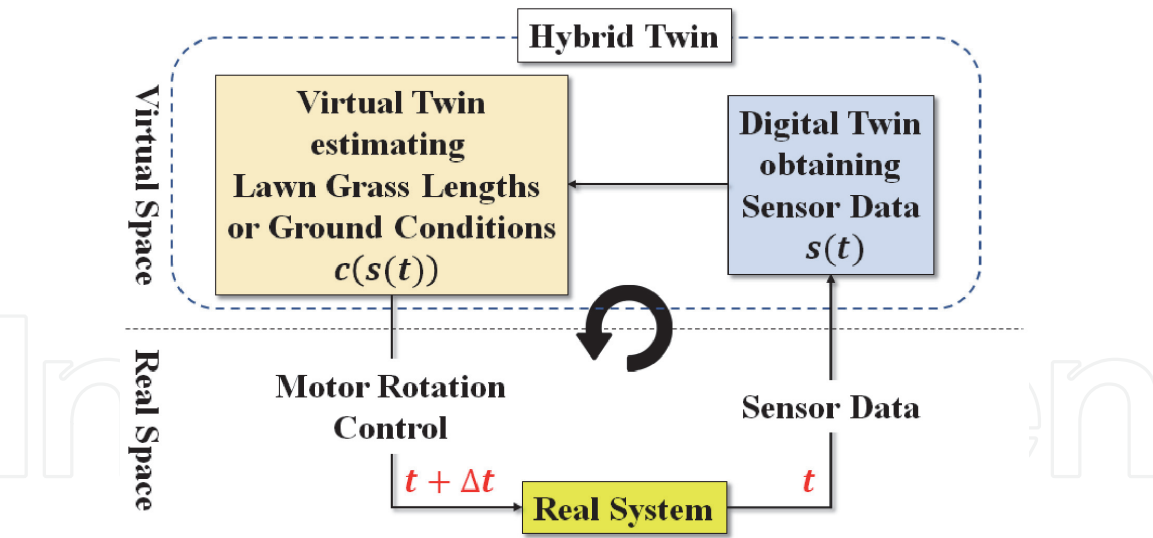


Figure 1.
Hybrid twin within work vehicles.

time (t), and then, estimation results $c(s(t))$ are obtained using AI algorithms. The control parameter set at time $(t + \Delta t)$ for motors is given to the real robo-mower. This loop is repeated during the operation of the robo-mower.

3. Work vehicles

This section describes the work vehicle used in the following discussions and experiments. A commercial robo-mower [8] is used as the experimental hardware for evaluating the proposed algorithms with the available fusion of sensors.

3.1 Robotic Lawn mower

The exterior and the backside of the robo-mower are shown in **Figure 2(a)**. The system configuration is shown in **Figure 2(b)**. It has two kinds of motor controllers,

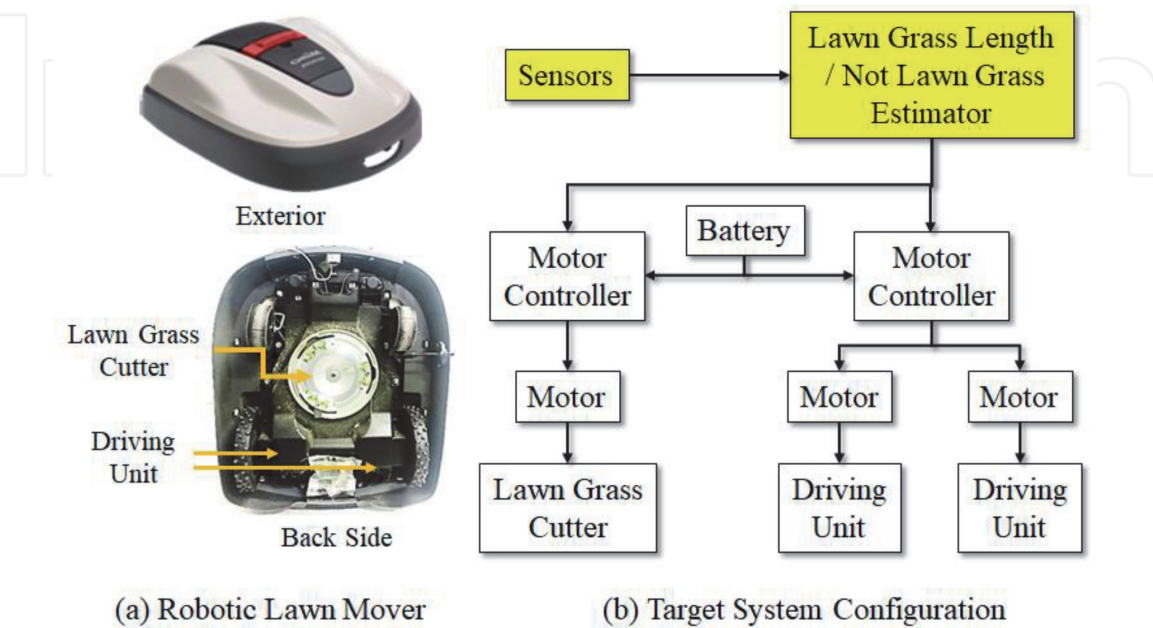


Figure 2.
Robotic Lawn mower and target system configuration.

one of which is used for controlling the lawn grass cutter and another is for its mobility. These controllers are powered by the same battery, and therefore, battery wastage depends on their usages. The subsystem, “Lawn Grass Length/Not Lawn Grass Estimator” is newly developed.

The robo-mower is used for experiments by attaching sensors, a single-board computer, a personal computer and peripheral devices on the robo-mower, as shown in **Figure 3**. All these devices are managed by an ROS (robot operating system) running on the personal computer. The robo-mower can be autonomously driven; however, it is controlled using a Bluetooth controlling device in the experiments to increase the accuracy of the experiments. A camera can be used as a sensor, but it is inadequate for the experiments due to its high cost of image processing software and hardware. In the experiments, it will be shown that no camera is needed for the required estimation.

3.2 Fusion of sensors

The sensors attached to the robo-mower shown in **Figure 3** are listed in **Table 1**. The robo-mower has originally been equipped with built-in sensors. The 9-axis

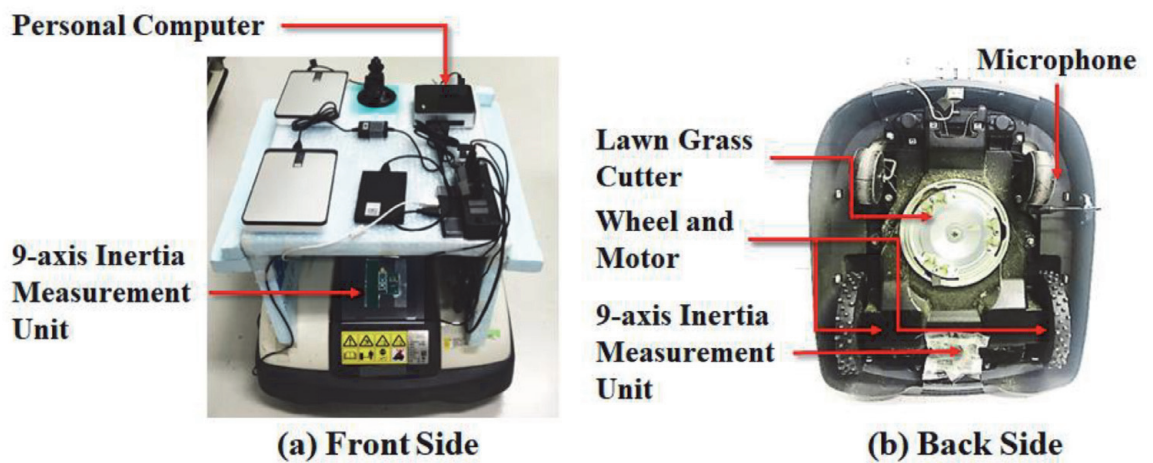


Figure 3.
Robotic Lawn mower with sensors and devices for development.

Sensors	Mounting positions	Measurements
9-axis Inertial Measurement Unit	Inside of the Robo-mower	Acceleration
		Angular Acceleration
	Surface of the Robo-mower	Acceleration
		Angular Acceleration
Built-In	Battery	Voltage
		Current
		Power
	Rotation of Grass Cutting Motor	
	Rotation of Traveling Motor	
	Horizontal/Vertical Acceleration	

Table 1.
Fusion of sensors.

inertial measurement units (IMUs), MPU-9250 [16], are attached inside and to the surface of the robo-mower to measure acceleration and angular acceleration. Six built-in sensors are available for measuring the corresponding parameters as shown in **Table 1**. Here the noise of sensors is negligibly small; however, outliers are excluded in the Digital Twin. The adequate fusion of these sensors is determined in each of the proposed algorithms, and this is verified in the experiments.

3.3 Estimation problem of Lawn grass lengths

The estimation problem of lawn grass lengths and ground conditions is defined below. The problem is to estimate lawn grass lengths in real time using sensor fusion data. The objective function is to increase the accuracy of estimation.

The Estimation Problem.

Input: set of available sensors, robo-mower's specifications, set of areas labeled. long lawn grass, short lawn grass, and without lawn. grasses, some of which are specified as test areas.

Output: fusion of sensors necessary for estimation and estimation results for test areas.

Objective Function: maximization of estimation accuracy.

4. AI-based approach

An AI-based approach is adopted for solving the estimation problem. The reason for this is that a combination of different types of sensor data should be handled, and the definition of long or short lawn grass is determined by the height of the lawn grass cutter from the ground. Moreover, a human operator estimates the length of a lawn grass based on sounds made by the lawn grass cutter while cutting grasses. Estimation using an AI-based approach is expected to be more efficient and accurate than estimation based on human judgment. The RF algorithm and SNN are adopted considering the execution speed in real-world applications.

4.1 Random Forest algorithm

The RF algorithm, a machine learning algorithm, originates from Breiman [17], and recently, its deep version has also been proposed [18]. This algorithm is used for classification, regression or clustering, etc. and is a type of ensemble algorithm using a set of decision trees as weak learners to avoid over-fitting and to maintain its high generalization performance. It is fast and achieves a comparatively high performance. According to the study [18], the deep RF algorithm achieves better results in specific applications while performing nearly as well in other wide applications. In the following, a specific RF algorithm is developed.

An RF algorithm consists of a given number of binary decision trees. The training and inference phases of the algorithm are shown in **Figure 4(a)** and **(b)**, respectively. In the configuration of binary decision trees, a set of training data sampled from input data is given to each of the binary decision trees. Then, the binary decision trees are constructed, as shown in **Figure 5**.

The data consist of the followings:

$\{n_i\}$ ($i = 1, 2, \dots, p$): input data for classification, regression or clustering, etc.

$\{x_i\}$ ($i = 1, 2, \dots, q$): features for classifying input data $\{n_i\}$.

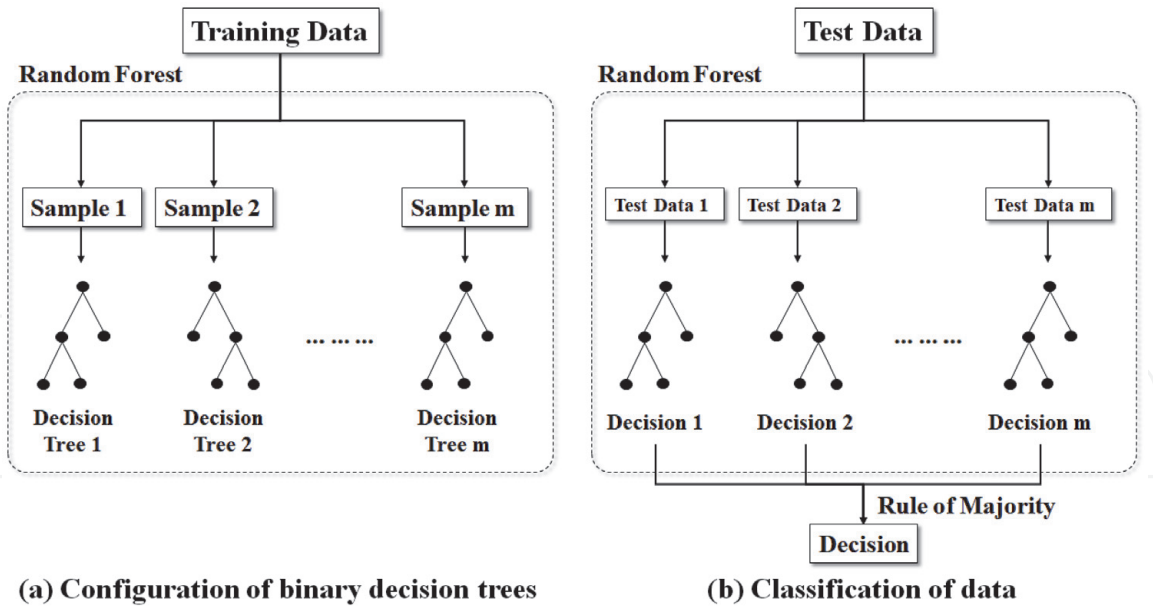


Figure 4.
Random forest algorithm.

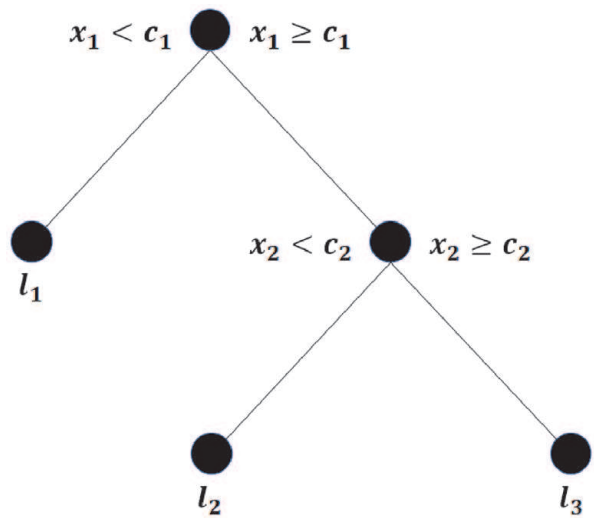


Figure 5.
Construction of binary decision tree.

4.1.1 Configuration of binary decision tree

An example of a binary decision tree is shown in Figure 5. In the root node, the input data are divided into two subsets using the conditions, $x_1 < c_1$ and $x_1 \geq c_1$. If the data satisfy the condition $x_1 < c_1$, the data are classified into the class l_1 as shown in this figure. When all data are classified into the corresponding classes (that is, leaves), the binary decision tree is completed. Here, for example, a classification and regression tree algorithm is used for classification, and the objective function is Gini's diversity index [18]. All parameters in binary decision trees are used in the classification phase.

4.1.2 Classification of data

For example, Bagging, an ensemble algorithm, is used for data classification. In this case, data that should be classified are distributed to all binary decision trees, and the decision of each binary decision tree is obtained. The final decision, that is,

the class to which the data belong, is determined on the basis of the majority rule. This process is faster if the binary decision trees are executed in parallel, and the decision quality is higher than if only one binary decision tree is used. As a 1D simulator, or a Virtual Twin, processing time for estimating the target area is essential.

4.2 Shallow neural network

As another AI-based approach, a SNN is used. “Shallow” means it has only one hidden layer, and this is expected to fasten processing. The estimation performance of SNN is compared with that of the RF algorithm.

4.2.1 Design of shallow neural network

The deep neural network performs well in image recognition, and it has been used in autonomous driving of automobiles, appearance inspection, image recognition of robots, and other applications. However, it requires a large amount of training data, and accordingly, a huge amount of processing time is needed in network training. In addition, the inference should also be executed on a GPU machine. On the contrary, the size of signal data is not so big because they are time-series, and no deep neural network may be needed for its recognition. For example, the on-line hammering sound inspection based on the simplest neural network with no hidden layers, a support vector machine, achieves more than 99% accuracy within a short time [19].

In the following, an SNN, as shown in **Figure 6**, is constructed. Here, the number of hidden layers is only one, and this layer has ten neurons. The number of neurons in the input layer equals the size of input signal length or statistical features such as the maximum value, minimum value, average value, median value, standard deviation value, peakedness value, and skewness value, all of which are obtained from the input sensor signal. The number of neurons in the output layer is two, that is, areas with or without lawn grasses, or with short or long lawn grasses. A hyperbolic tangent activation function and a softmax function are incorporated into the output layer.

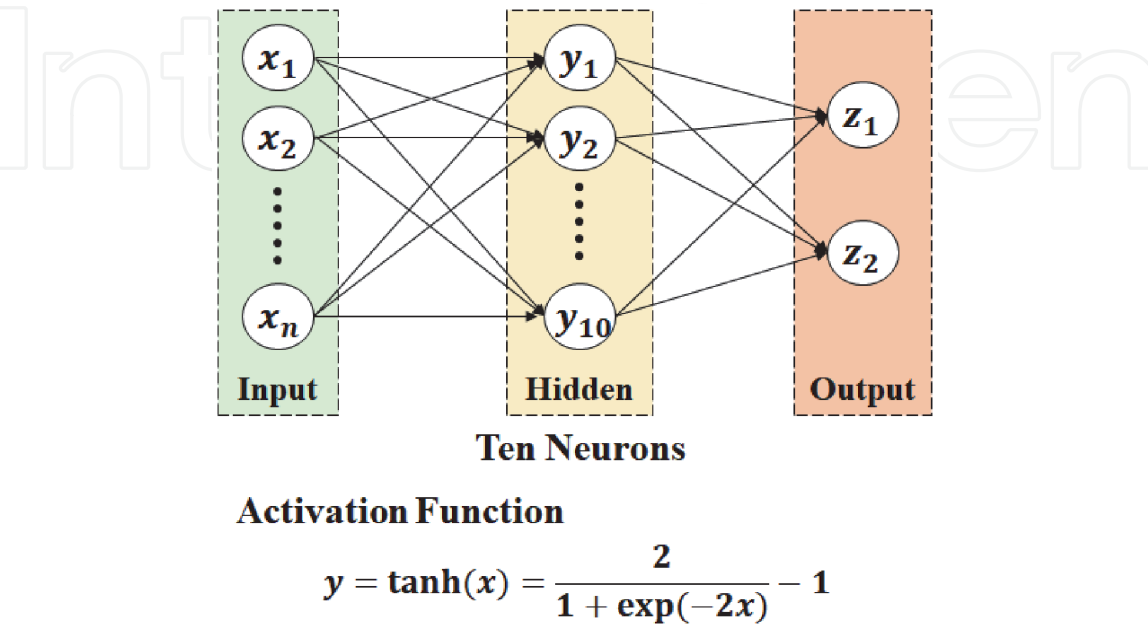


Figure 6.
Shallow neural network configuration.

4.2.2 Application of shallow neural network

In the following, two SNN models are developed with a cascade connection. The first model estimates the target areas that are with or without lawn grasses because this can be distinguished according to the acceleration of the robo-mower operating in a corresponding area. The second model estimates the height of the lawn grass as long or short. It is expected that this might be determined by checking the current of the motor depending on the load on the cutting blade. These two models are connected in series, that is, a cascade configuration. Furthermore, these SNNs are trained independently.

5. Experimental results

This section focuses on the experiments and evaluations of the RF algorithm and SNN on real-world sensing data.

5.1 Experiments on RF algorithm

5.1.1 Measurement data

The data measured by the sensors are obtained by driving the robo-mower on a field with long lawn grasses and short lawn grasses as well as without lawn grasses. The actual remote-controlled robo-mower is shown in **Figure 7**. The remote-control system through Bluetooth communication is incorporated in the robo-mower by mounting a mini-PC and running an ROS on it. The mini-PC can also handle the collected sensor data.



Figure 7.
Remote-controlled driving of Robo-mower.

The collected data are manually categorized into long and short lawn grass datasets according to the height of the grass cutter from the ground. If the length of lawn grasses exceeds the height of the grass cutter, the lawn grass length is defined as long and otherwise, short. When the heights of lawn grasses and grass cutter are equal, a human operator determines whether the lawn grass is long or short according to the operating sound of the grass cutter. The measurement data are collected within a total time of 2.3 h. All data are collected on flatlands on sunny days.

5.1.2 Features for classifying data

Statistical features of input data $\{x_i\}$ ($i = 1, 2, \dots, q$) for classification are calculated, including (i) maximum value, (ii) minimum value, (iii) average value, (iv) median value, (v) standard deviation value, (vi) kurtosis value, and (vii) skewness value. The values of seven feature types are normalized into the interval $[-1,1]$. These features are used in configuring binary decision trees, and they are calculated for each time frame obtained approximately every 3.2 s over 2.3-h measurement data. The details of the collected data are shown in **Table 2**. Even if the total time for data collection is less than 2.3 h because of, for example, some issues with measurement devices, the obtained data are used.

In the experiments, a subset of time frames obtained from each field data is used for configuring the binary decision trees, and the completed forest is applied to the remaining test data. Then, classification performance of the RF algorithm is evaluated.

The number of time frames (training data) used to configure the binary decision trees in each group is chosen at random from the measurement data. The remaining time frames are used as test data for evaluating the RF’s performance. These are shown in **Table 3**.

5.1.3 Evaluation criteria

Each of the time frame data has its label, that is, long lawn grasses, short lawn grasses, and not lawn grasses, and the estimation can be verified. This process

Groups	Number of time frames
Long Lawn Grasses	2,356
Short Lawn Grasses	1,575
Not Lawn Grasses	3,374
Total	7,305

Table 2.
Specifications of measurement data for evaluating RF algorithm.

Groups	Number of time frames	
	For training	For testing
Long Lawn Grasses	1,686	670
Short Lawn Grasses	904	671
Not Lawn Grasses	2,705	669
Total	5,295	2,010

Table 3.
Number of time frames for training and testing RF algorithm.

consists of two stages. The first stage is used to estimate whether an area is with or without lawn grasses. In the second stage, an area is further estimated whether it has long or short lawn grasses when it is estimated to have lawn grasses. In the testing, four kinds of evaluation criteria are used. These are defined below [20].

		Actuals	
		Positive	Negative
Predictions	Positive	TP	FP
	Negative	FN	TN

1. Accuracy

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{1}$$

2. Precision

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

3. Recall

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

4. F-Measure

$$F - Measure = \frac{2 * Precision * Recall}{Precision + Recall} \tag{4}$$

where *TP*, *TN*, *FP*, and *FN* denote “True Positive,” “True Negative,” “False Positive,” and “False Negative,” respectively.

5.1.4 Evaluation results

Seven combinations of sensor data used are shown in **Table 4**. These combinations cover all possible cases. Using the measurement data from **C1** to **C7**, the best combination of sensor data is determined based on the above-mentioned evaluation criteria.

The procedure of experiments is as follows.

1. Select the sensor data corresponding to the cases shown in **Table 4** collected in three ground conditions, that is, “Long Lawn Grasses,” “Short Lawn Grasses,” and “Not Lawn Grasses.”

Determine the subset of sensor data (1) and partition it to configure the binary decision trees and to test the RF algorithm according to the number of the time frames shown in **Table 3**.

2. Configure the binary decision trees.
3. Evaluate the performances of the RF algorithm based on the evaluation criteria.

The number of binary decision trees, that is, the size of the forest is set to 1,000. Each binary decision tree is configured using the seven features mentioned in 5.1.2 until each leaf coincides with one of three ground conditions. An example of an actually constructed binary decision tree is shown in **Figure 8**. Here, the feature, median value, obtained from built-in vertical angle sensor data with its threshold 370.75, is used for classifying the input data on the root node. The class

Sensors	Mounting positions	Measurements	C1	C2	C3	C4	C5	C6	C7
9-axis Inertial Measurement Unit	Inside of Body	Acceleration	√	√		√	√	√	√
		Angular Acceleration	√	√		√	√	√	√
	Surface of Body	Acceleration		√			√		√
		Angular Acceleration		√			√		√
Built-In	Battery	Voltage				√	√	√	√
		Current				√	√	√	√
		Power				√	√	√	√
	Rotation of Grass Cutting Motor					√	√	√	√
						√	√	√	√
						√	√	√	√

Table 4. Combinations of sensor data.

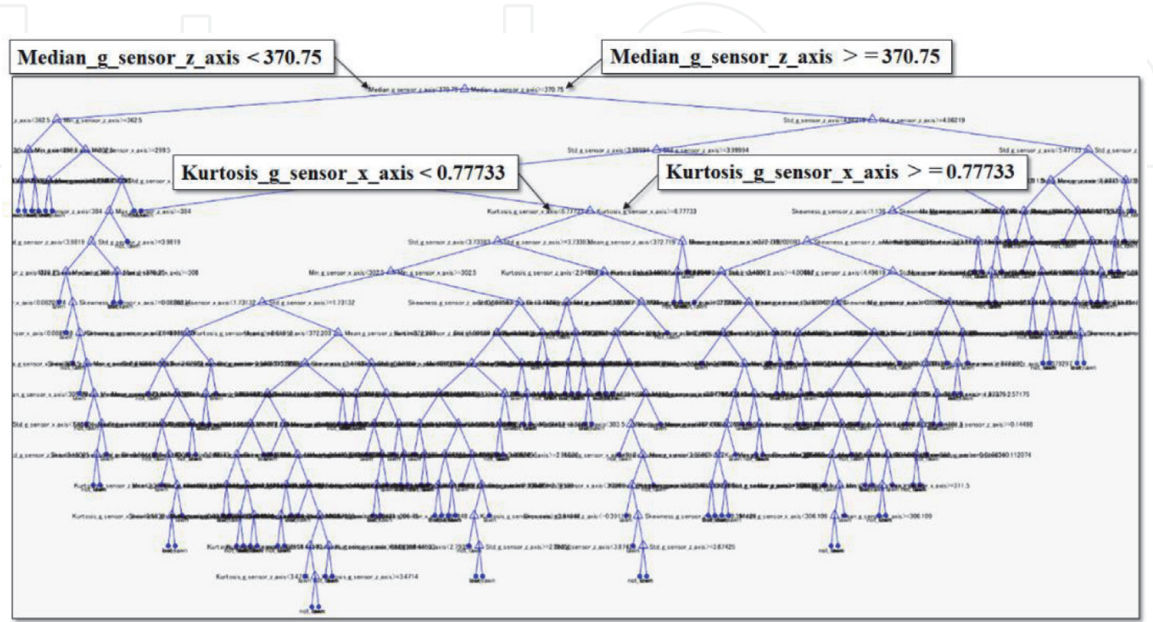


Figure 8. Example of binary decision tree in RF algorithm.

“TreeBagger” included in “Statics and Machine Learning Toolbox” in MATLAB [21] is used for implementing the RF algorithm. The processing time for configuring 1,000 binary decision trees is less than ten minutes on a PC with the standard performances. The completed forest is applied to the testing data whose size is approximately 700 in each of the three ground conditions shown in **Table 3**. The estimating time is negligibly small, and this is no issue in the actual Hybrid Twin approach. **Table 5** shows the performance of the algorithm according to different sensor data combinations. Seven cases are evaluated with respect to the measurement criteria in each ground condition. The most important performance is the accuracy, and it increases when the built-in sensor data are used. Particularly, **C6** and **C7**, excluding the built-in horizontal or vertical angle sensor, have higher accuracy. It seems reasonable that the battery status and motor rotation conditions contribute to higher performance because the rotation of the motor becomes high when it encounters long lawn grasses. On the other hand, the load on both the grass cutting motor and traveling motor is reduced when the robo-mower travels on a ground without lawn grasses. From the evaluation results, **C6** is desirable among seven cases. The reason is that

1. the accuracy is high, with a difference of only 0.1 points from maximum 92.28%,
2. the recall ratio of Short Lawn Grasses, 87.08%, is the highest.

Especially, the low recall ratio of Short Lawn Grasses means that the probability of incorrectly recognizing short lawn grasses as long lawn grasses or a ground other than lawn grasses is high. Then, the traveling speed of the robo-mower is reduced, and the rotation of the grass cutting motor is increased. This would increase the working time and waste electric energy. Moreover, when a short lawn grass area is incorrectly classified as a ground without lawn grasses, the robo-mower will not move on the areas and will not cut lawn grasses. Therefore, it would be concluded that **C6** is the best combination in this evaluation results. Sensor data, including the acceleration and angular acceleration values obtained using the 9-axis IMU attached inside the robo-mower; the voltage, current, and power of the battery; the rotation

Combinations of sensor data		C1	C2	C3	C4	C5	C6	C7
Accuracy		75.84	77.52	86.32	90.92	91.58	92.18	92.28
Long Lawn Grasses	Precision	77.06	70.70	92.68	92.60	91.46	92.66	91.04
	Recall	73.02	67.70	93.46	93.60	93.64	93.44	93.56
	F-measure	74.99	69.17	93.07	93.10	92.54	93.05	92.28
Short Lawn Grasses	Precision	72.66	73.84	87.30	90.48	92.12	89.40	92.04
	Recall	60.88	68.28	70.06	82.00	83.26	87.08	85.58
	F-measure	66.25	70.95	77.73	86.03	87.47	88.22	88.69
Not Lawn Grasses	Precision	77.06	86.34	80.24	89.70	91.34	94.40	93.72
	Recall	93.60	96.60	95.46	97.20	97.90	96.08	97.70
	F-measure	84.53	91.18	87.19	93.30	94.51	95.23	95.67
Average of Precision, Recalls, and F-measures		75.56	77.20	86.35	90.89	91.58	92.17	92.25

Table 5.
Evaluation results for sensor fusions.

speed of the grass cutting and traveling motors obtained using built-in sensors, are used in C6.

5.2 Experiments for SNN

5.2.1 Available sensors

The built-in sensors attached to the robo-mower are listed in **Table 1**. Six built-in sensors are available for measuring the corresponding parameters shown in **Table 1**. The objectives of the experiments are first to evaluate the accuracy of lawn grass height estimation and ground condition estimation and, second, to compare the SNN’s results with those of the RF algorithm.

5.2.2 Measurement data

The data measured by sensors are collected while driving the robo-mower on a field with long and short lawn grasses as well as without lawn grasses. The three types of lawn grasses are shown in **Figure 9(a)–(c)** are used for the experiments. In each of these cases, the long lawn grass case and short lawn grass case are performed by adjusting the lawn grass cutting blade height from the ground. Similarly, several ground conditions without lawn grasses are adopted as shown in **Figure 9(d)–(f)**, which are asphalt, gravel, and stone pavement, respectively.

The collected data are categorized into three groups. The first group is for long lawn grasses, that is, the height of lawn grass is larger than the specified one. The second group is for the short lawn grasses, that is, the lawn grass is shorter than or equal to the specified one. The third group is the field without lawn grasses, that is, dirt, gravel, stone pavement, tiled, asphalt, or concrete fields. The measurement data are collected for a total driving time of 2.3 h for each group with various fields.

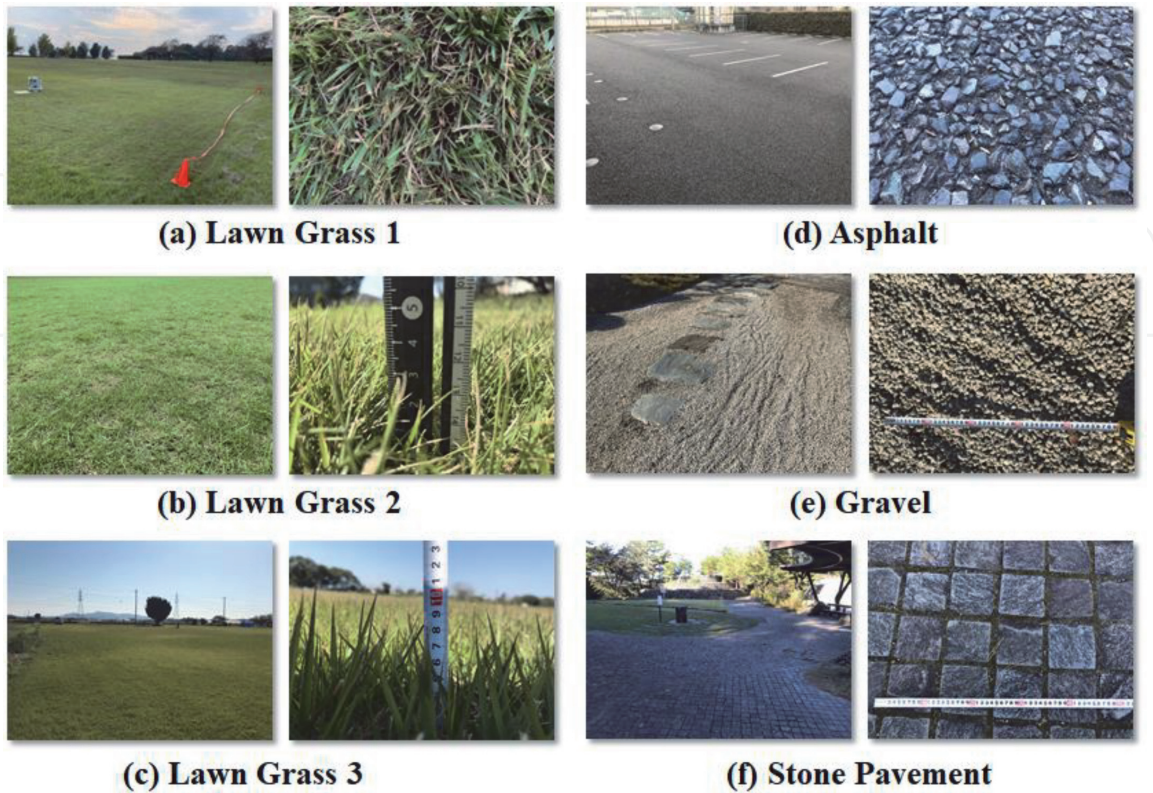


Figure 9.
Variations of target areas.

5.2.3 Fusion of sensors for SNN

The fusions of sensors for two SNNs SNN1 and SNN2 are experimentally determined. Ten types of sensors, as shown in **Table 1**. As **Figure 10** shows, SNN1 and SNN2 are cascade connected, where SNN1 estimates if a target area is with or without lawn grasses, and SNN2 provides the result that an area is with long/short lawn grasses. The basic idea behind this configuration is that sensor fusion may differ with these two types of estimations.

For SNN1, the Horizontal/Vertical Acceleration sensor seems adequate, and the combinations of x , y , and z -axis sensor data are experimented with and evaluated on some datasets. The obtained accuracy is shown in **Figure 10**. For the estimation of lawn lengths, sensor fusion indicating the load of the lawn grass cutter seems effective. Therefore, the measurement data obtained from the combinations of the battery sensor and duty ratio given to the cutting motor are evaluated with some datasets. As shown in **Figure 10**, the multiplication of battery voltage and the duty ratio for the cutting motor achieves maximum accuracy. The differences are minor; however, this multiplication is probably the reason for showing the load of grass cutting motor.

As a result, the x -axis and z -axis values obtained from the Horizontal/Vertical Acceleration sensor are used as inputs for SNN1. Similarly, the multiplication of battery voltage and the duty ratio for the grass cutting motor, respectively, obtained from Battery Voltage and Rotation of Grass Cutting Motor are used as inputs for SNN2.

5.2.4 Features for classifying data

The input data features are (i) maximum value, (ii) minimum value, (iii) average value, (iv) median value, (v) standard deviation value, (vi) peakedness value, and (vii) skewness value. In signal recognition based on machine learning, some

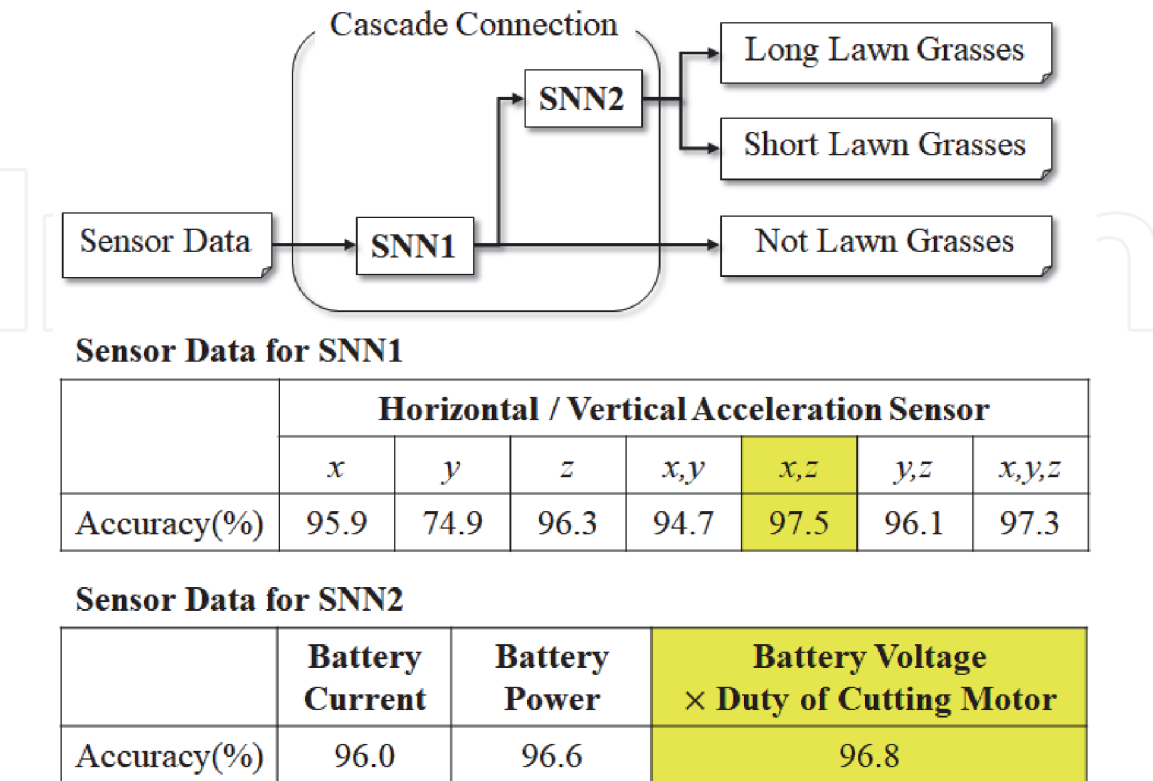


Figure 10.
Fusion of sensors for SNN.

features are typically extracted from input signal data during pre-processing. The obtained features are used as input data to a machine learning algorithm. Therefore, for each sensor data, seven neurons are needed in the input layer of SNN shown in **Figure 6**. In the experiments, these features are obtained from a time frame obtained every 3.2 s over 2.3-h measurement data.

The details of collected data are shown in **Table 6**. In the experiments, a subset of time frames from each field data is used for training the SNN, and the obtained model is tested on the remaining test data. The number of time frames used for testing is approximately 670 in each group. The remaining time frames are used for training, as shown in **Table 7**.

5.2.5 Shallow neural network construction

Two SNNs are constructed, and they are cascade connected. The first SNN1 estimates whether the ground is with or without lawn grasses using the horizontal or vertical acceleration sensor. These sensors are expected to measure the acceleration changes caused by the surface of the ground. There are seven feature types, as mentioned in 5.2.4, and 14 neurons in the input layer of SNN1. The second SNN2 estimates the lawn grass lengths using the product of the battery voltage and duty ratio of the motor control signal. This product value tends to vary according to the loads given to the cutting motor. The number of neurons in the input layer of SNN2 is seven. These networks are constructed using “Statistics and Machine Learning Toolbox” in MATLAB [22]. The specifications of the two SNNs are summarized in **Table 8**.

5.2.6 Evaluation results

In the following, two SNNs are first trained, and next, they are used to estimate lawn grass lengths or ground conditions. Their results are compared with those of the RF algorithm. The evaluations are repeated ten times, and their averages are

Groups	Number of time frames
Long Lawn Grasses	2,356
Short Lawn Grasses	1,574
Not Lawn Grasses	2,470
Total	6,400

Table 6.
Specifications of measurement data for evaluating SNN.

Groups	Number of time frames	
	For training	For testing
Long Lawn Grasses	1,686	670
Short Lawn Grasses	904	670
Not Lawn Grasses	1,801	669
Total	4,391	2,009

Table 7.
Number of time frames for training and testing SNN.

	Number of neurons			Used sensors
	Input	Hidden	Output	
SNN1 for Ground Condition	14	10	2	Horizontal or Vertical Acceleration Sensor
SNN2 for Long/Short Lawn Grasses	7	10	2	Battery Voltage & Control Signal's Duty

Table 8.
Configurations of two SNNs.

	Estimation	SNN	RF	Diff.
Precision	Long Lawn Grasses	92.3	94.1	-1.8
	Short Lawn Grasses	96.2	92.8	+3.4
	Not Lawn Grasses	95.8	95.5	+0.3
Recall	Long Lawn Grasses	97.6	94.1	+3.5
	Short Lawn Grasses	90.9	92.4	-1.5
	Not Lawn Grasses	96.3	95.9	+0.4
F-Measure	Long Lawn Grasses	94.9	94.1	+0.8
	Short Lawn Grasses	93.5	92.6	+0.9
	Not Lawn Grasses	96.0	94.1	+0.3
Accuracy		94.8	94.1	+0.7

Table 9.
Evaluation of SNN.

used because the set of training data is randomly selected from the set of time frames, as shown in **Table 7**.

The evaluation results are shown in **Table 9**. The evaluation results corresponding to the evaluation measurements defined in 5.1.3 are shown in this table. The accuracy of the SNN outperforms that of the RF algorithm on average, and the differences in evaluation criteria of each estimation are shown in the “Diff” column in **Table 9**. Except for the Precision of “Long Lawn Grasses” and the Recall of “Short Lawn Grasses,” the differences are positive. However, the differences are not as large in all estimations, and the required estimation time is negligible.

6. Evaluations of SNN against sensor data stream

As stated in 5.2.6, the SNNs outperforms the RF algorithm, and the estimation system based on the SNNs is implemented on a Raspberry Pi assuming an actual ECU. The processes Lawn Grass Length/Not Lawn Grass Estimator are shown in **Figure 11**. The sensor data streams are sent in a serial format and are received and saved in the memory of the Raspberry Pi. When the required data size is reached, a set of data is preprocessed. Seven features mentioned in 5.2.4 are extracted according to the sensors, including Horizontal/Vertical Acceleration and Battery Voltage times Duty of Cutting Motor, as shown in **Figure 10**. Then, the Lawn Length/Not Lawn Grass Estimator based on the SNNs estimates that a target area is with Long Lawn Grasses, Short Lawn Grasses, or without Lawn Grasses. Finally, the estimation result is sent to the motor controllers, as shown in **Figure 2(b)**.

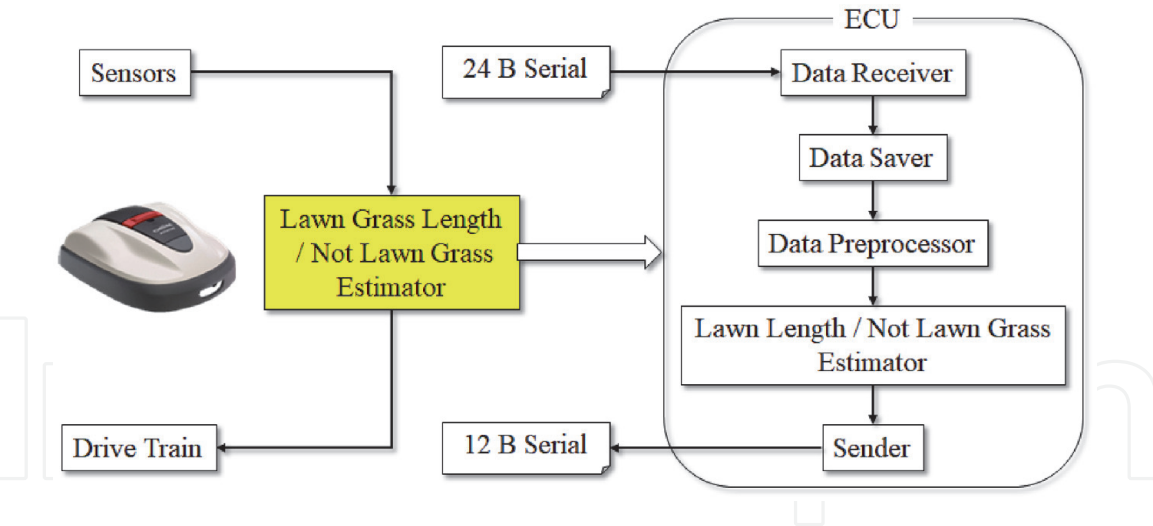


Figure 11.
Implementation of estimator on ECU.

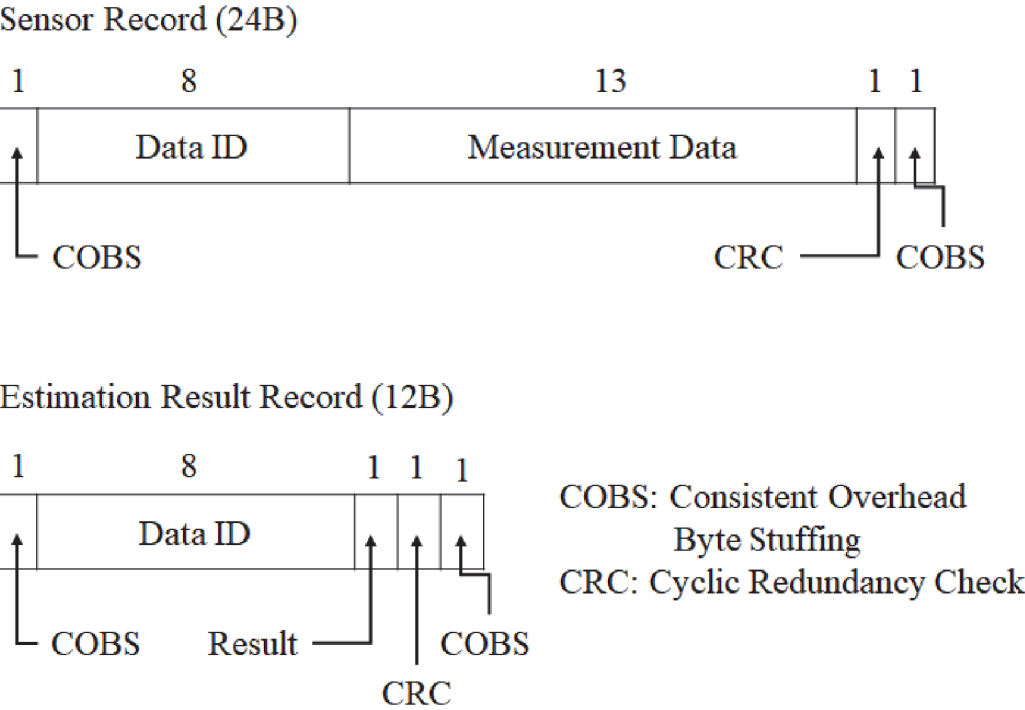


Figure 12.
Frame Design of Sensor and Estimation Result Records.

Figure 12 shows the frames of sensor records and estimation result record. The sensor record has 24 bytes, and it is framed using COBS (Consistent Overhead Byte Stuffing) [23]. This frame has 8-bit Data ID, 13-bit Measurement Data (floating point value), and a 1-bit CRC (cyclic redundancy check) [24] for error detection. The frame of estimation result record consists of COBS, CRC, Data ID, and the estimation result, which is a 1-byte integer (1, 2, or 3) representing the three kinds of estimations.

The sensor data records are obtained every 0.1 s. The 32 records, that is, the 3.2-s measurement data are handled in the SNN training, as shown in **Figure 13(a)**. The robo-mower travels with a speed of 0.55 m/s on average [8], and it moves 1.76 m before the 3.2-s sensor data are obtained. Furthermore, if the total processing time for data I/O, motor control, and wheel driving are assumed to be 2 s, the robo-mower will travel 2.86 m if the estimation waits for a 3.2-s data stream. This distance is too large when the robo-mover reaches the boundary between areas with and without lawn grasses. Therefore, as shown in **Figure 13(b)**, the sensor data stream should be treated

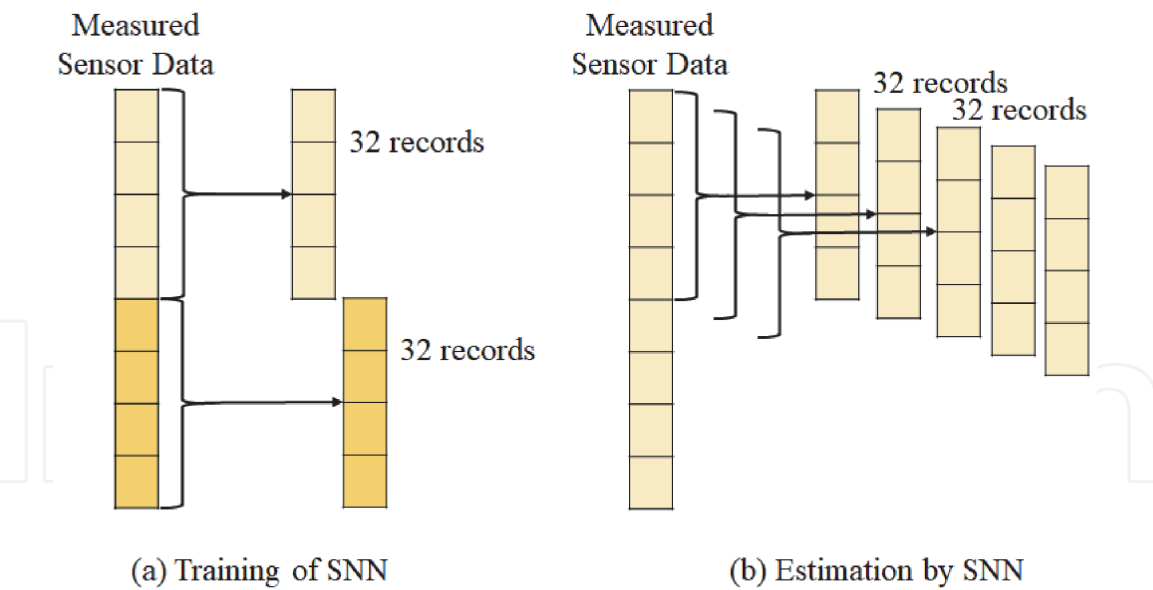


Figure 13.
Pipeline processing of Lawn length estimation.

	Estimation	Pipeline	Non-Pipeline	Diff.
Precision	Long Lawn Grasses	98.9	97.2	+1.7
	Short Lawn Grasses	88.7	90.8	-2.1
	Not Lawn Grasses	94.7	97.4	-2.7
Recall	Long Lawn Grasses	97.6	92.7	+4.9
	Short Lawn Grasses	95.0	96.7	-1.7
	Not Lawn Grasses	89.3	95.5	-6.2
F-Measure	Long Lawn Grasses	98.3	94.9	+3.4
	Short Lawn Grasses	91.7	92.8	-1.1
	Not Lawn Grasses	91.9	95.4	-3.5
Accuracy		94.0	95.0	-1.0

Table 10.
Accuracies between pipeline and non-pipeline processes.

as a pipeline. This means that the estimation is performed every 2.1 s (i.e., 0.1 + 2), resulting in 1.16-m traveling, which is not a significant issue in the real world. The other cause for the estimation delay is the acceleration of other related processes.

Another problem to consider when using the robo-mower in the real world is estimation accuracy. This is because the SNNs are trained with each 3.2-s data frame of sensor data, as shown in **Figure 13(a)**, ensuring that there are no overlaps in the consecutive dataset. The data streams in the pipeline processing have 3.1-s overlap because they are obtained every 0.1 s. The accuracy comparison result is shown in **Table 10**, and the decrement is 1.0 points. Therefore, the estimation accuracy in the pipeline process is not a major issue.

7. Conclusions

The workload estimation methods for autonomous driving of work vehicles are proposed and evaluated. A commercial electric robo-mower is used for the

experiments. Specifically, the task to recognize ground conditions, including grounds with long lawn grasses, short lawn grasses, or no lawn grasses, is handled by analyzing data obtained from sensors attached to the robo-mower. Two AI-based algorithms, namely, an RF algorithm and a SNN are proposed. A sensor fusion problem is defined and solved to determine the best combination of sensor data from ten different sensor types. The RF algorithm consisting of 1,000 decision trees and the SNN with only one hidden layer are implemented and evaluated on observation data obtained from various grass cutting field experiments. The RF algorithm achieves 92.3% correct estimation ratio on sensor fusion data in several experiments, while the SNN achieves 95.0%. Furthermore, the accuracy of the SNN is 94.0% in experiments where sensing data are continuously collected as a data stream in real time while the robo-mower is operating. Presently, the proposed estimation system is being developed by integrating two motor control systems into a robo-mower, one for grass cutting and the other for the robot's mobility.

Acknowledgements


The authors express their sincere appreciation to the engineers at Life Creation Center, Honda R&D Co., Ltd., for their technical and financial supports.

Author details

Yoichi Shiraishi*, Haohao Zhang and Kazuhiro Motegi
Graduate School of Mechanical Science and Technology, Gunma University, Ohta,
Gunma, Japan

*Address all correspondence to: yoichi.siraisi@gunma-u.ac.jp

IntechOpen

© 2021 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/3.0>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. 

References

- [1] Zushida K, Zhang H, Shimamura H, Motegi K, Shiraishi Y: Application and Analysis of Random Forest Algorithm for estimating Lawn Grass Lengths in Robotic Lawn Mower. *International Journal of Mechanical Engineering and Applications*. 2021;9:1:6–14. DOI: 10.11648/j.ijmea.20210901.12
- [2] Kojima M, Shimamura H, Motegi K, Shiraishi Y: Comparison of Shallow Neural Network with Random Forest Algorithm in Estimating Lawn Grass Lengths for Robotic Lawn Mowers. In: *Proceedings of the SICE Annual Conference 2020*; 23–26 September 2020; Thailand; 2020. p. 1634–1639
- [3] Ootake K, Shimamura H, Motegi K, Shiraishi Y: A Travelling Simulator for Robotic Lawn Mowers based on Particle Filter Approach. In: *Proceedings of the SICE Annual Conference 2020*; 23–26 September 2020; Thailand; 2020. p. 1640–1645
- [4] Aoki A, Shimamura H, Nakata M, Motegi K, Shiraishi Y: Stuck State Avoidance in Work Vehicle's Control with Hybrid Twin Framework. *Universal Journal of Control and Automation*. 2020;8:2:23–31. DOI: 10.13189/ujca.2020.080201.
- [5] Zhang H, Watanabe K, Motegi K, Shiraishi Y: ROS Based Framework for Autonomous Driving of AGVs. In: *Proceedings of ICMEMIS*; December 2019; 4–6 December 2019; Japan; 2019. IPS6–04
- [6] Marc-Oliver S, Max L, Agathe K, Florian K, Breitner, M H: Autonomous Unmanned Ground Vehicles for Urban Logistics: Optimization of Last Mile Delivery Operations. In: *Proceedings of the 52nd Hawaii International Conference on System Sciences*; 2019;p.1538–1547
- [7] Alatisse M B, Hancke G P: A Review on Challenges of Autonomous Mobile Robot and Sensor Fusion Methods. *IEEE Access*. 2020;8:39830–39846. DOI: 10.1109/ACCESS.2020.2975643
- [8] Kawakami T, Kobayashi H, Yamamura M, Maruyama S: Development of Robotic Lawnmower Miimo. *Honda R&D Technical Review*. 2013;25:2:54–59. September 2013 Available from: <https://www.hondarandd.jp/point.php?pid=968&lang=en>.
- [9] Khaleghi B, Khamis A, Karray F O, Razavi S N: Multisensor data fusion: A review of the state-of-the-art. *Information Fusion*, 2013;14: 28–44
- [10] Alam F, Mehmood R, Katib I, Albogam N N, Alebeshr A: Data Fusion and IoT for Smart Ubiquitous Environments: A Survey. *Special Section on Trends and Advances for Ambient Intelligence with Internet of Things (IoT) Systems*. 2017;5:9533–9554
- [11] Takami G, Tokuoka M, Goto H, Nozaka Y: Machine Learning Applied to Sensor Data Analysis. *Yokogawa Technical Report English Edition*. 2016; 59:1:27–30
- [12] Alonso S, Perez D, Moran A, Fuertes J J, Diaz I, Dominguez M: A Deep Learning Approach for Fusing Sensor Data from Screw Compressors. *Sensors* 2019. 2019;19:13:2868 available from: <https://doi.org/10.3390/s19132868>
- [13] Li C, Sanchez R V, Zurita G, Corrada M, Cabrera D: Fault Diagnosis for Rotating Machinery Using Vibration Measurement Deep Statistical Feature Learning. *Sensors*. 2016;16:895. doi: 10.3390/s16060895
- [14] Chinesta F, Cueto E, Abisset-Chavanne E, Virtual L D J: Digital and Hybrid Twins: A New Paradigm in Data-Based Engineering and Engineered Data. *Archives of Computational*

Methods in Engineering. 2018; DOI:
10.1007/s11831-018-9301-4

[15] El Saddik A: Digital Twins: The
Convergence of Multimedia
Technologies. IEEE Multimedia. 2018;
25:2:87–92

[16] MPU-9250. available from: [https://
www.invensense.com/produc
ts/motion-tracking/9-axis/mpu-9250/](https://www.invensense.com/products/motion-tracking/9-axis/mpu-9250/)

[17] Breiman L: Random Forests
Machine Learning. 2001;45:5–32

[18] Zhou Z, Feng J: Deep Forest:
Towards An Alternative to Deep Neural
Networks. In: Proceedings of the
Twenty-Sixth International Joint
Conference on Artificial Intelligence.
2017;p.3553–3559 doi:10.24963/
ijcai.2017/497.

[19] Oka D, Hiroshan Lakmal Balage D,
Motegi K, Kobayashi Y, Shiraishi Y: A
Combination of Support Vector
Machine and Heuristics in On-line Non-
Destructive Inspection System. In:
Proceedings of the International
Conference on Machine Learning and
Machine Intelligence (MLMI).
September 2018; Vietnam; 2018.
CM0007 archived in IEEE Xplore
Digital Library

[20] H. Witten I, Eibe Frank E, A. Hall
M: Data Mining: Practical Machine
Learning Tools and Techniques: Morgan
Kaufmann; 2011

[21] TreeBagger, Math Works. available
from: [https://jp.mathworks.com/ help/
stats/treebagger.html?lang=en](https://jp.mathworks.com/help/stats/treebagger.html?lang=en).

[22] Statistics and Machine Learning
Toolbox, MathWorks. available from:
[https://www.mathworks.com/products/
statistics.html#machine-learning](https://www.mathworks.com/products/statistics.html#machine-learning)

[23] Consistent Overhead Byte Stuffing.
available from: [https://en.wikipedia.org/
wiki/Consistent_Overhead_Byte_
Stuffing](https://en.wikipedia.org/wiki/Consistent_Overhead_Byte_Stuffing)

[24] Cyclic redundancy check. available
from: [https://en.wikipedia.org/wiki/Cyc
lic_redundancy_check](https://en.wikipedia.org/wiki/Cyclic_redundancy_check)