Predicting bucket-filling control actions of a wheel-loader operator using a neural network ensemble*

Siddharth Dadhich[1], Fredrik Sandin[2] and Ulf Bodin[2]

Abstract— **Automatic bucket filling is an open problem since three decades. In this paper, we address this problem with supervised machine learning using data collected from manual operation. The range-normalized actuations of lift joystick, tilt joystick and throttle pedal are predicted using information from sensors on the machine and the prediction errors are quantified. We apply linear regression, k-nearest neighbors, neural networks, regression trees and ensemble methods and find that an ensemble of neural networks results in the most accurate predictions. The prediction root-mean-square-error (RMSE) of the lift action exceeds that of the tilt and throttle actions, and we obtain an RMSE below 0.2 for complete bucket fillings after training with as little as 135 bucket filling examples.**

I. INTRODUCTION

Wheel-loaders are multi-purpose machines fitted with different kinds of attachments such as buckets and forks. In construction industry, wheel-loaders are mostly used with a bucket to transport materials such as soil, gravel and rock. Autonomous excavation and automatic bucket filling is difficult and is an open problem for three decades [1].

Earlier works on automated bucket filling are based on trajectory-planning in compliance with measured excavation forces on the bucket [2]. The forces between the material and the bucket are difficult to measure and model. Trained human operators perform better than models when filling the bucket. Therefore, statistical modeling appear suitable for this problem [3].

The aim of this work is to evaluate the use of machine learning algorithms for predicting an operator's control actions during bucket filling. Application of machine learning to automate the bucket filling process is feasible in principle [4] and can lead to flexible solutions because the model can be adapted to a new machine, material or environmental condition by training with an appropriate dataset and/or reinforcement learning. In this paper, supervised learning is used to train models for predicting the operator control actions: Lift joystick, tilt joystick and throttle pedal. The training data is recorded during a controlled experiment with an expert driver filling buckets of medium course gravel with a wheel-loader, as shown in Fig. 1.

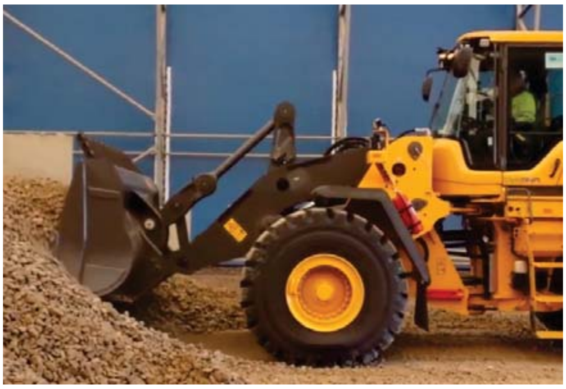

Fig. 1. Experiment used to generate data for training and testing of machine-learning models. An expert operator fill buckets of medium course gravel with a Volvo L180 wheel-loader with custom software.

Operators use vision, vibration, tactile and vestibular feedback to operate the machine and fill the bucket. In this work, we investigate how accurately the control actions of the operator can be automatically predicted using data from standard sensors in wheel-loaders, such as the pressure transducers in the hydraulics. We assume that the control actions of an operator can be predicted with statistical models that are fitted to recorded sensor readings and control actions from several bucket fillings.

The prediction of control actions is a time-series regression problem, which appears in many other contexts like rainfall forecasts [5], battery state estimation [6], building energy prediction [7], [8], household electricity forecasting [9], and the control of HVAC systems [10]. However, the bucket filling problem involves a human operator directly in the control loop, and the dynamical processes of interest have relatively short timescales.

Traditional control theory is not easy to apply to the bucket filling problem. For example, it is not clear which variables are to be controlled. A major problem is to model the interaction between the bucket and the excavated material. Force (impedance/admittance) control is used in problems where a robot (or robotic-arm) interacts with its environment [11]. A problem in excavation is that the pile is rigid and that the reaction forces from the pile during bucket filling are sometimes exceeding the applied forces [12]. However, Dobson [13] recently presented an admittance controller for automatic bucket-filling. That solution requires experimental tuning of the admittance based PID controller and has provided promising results.

The main contribution of this paper is an alternative machine-learning approach to automate the bucket filling process, which complements the previous work with an

*This work is supported by the Swedish Innovation Agency, VINNOVA.

[1] S. Dadhich is an PhD candidate at the Department of Electrical, Computer Science and Space Engineering, Luleå University of Technology, 971 87, Luleå, Sweden.siddharth.dadhich@ltu.se

[2] F. Sandin and U. Bodin are with the faculty at the Department of Computer Science, Electrical and Space Engineering, Luleå University of Technology, 97187, Luleå, Sweden.

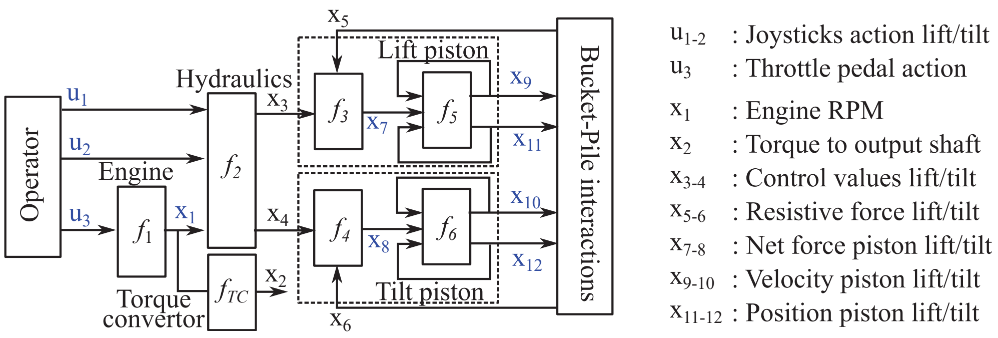
Operator
u1
u2
u3
Engine
f1
x1
Hydraulics
f2
x3
x4
Torque convertor
fTC
x2
x5
Lift piston
f3
x7
f5
x9
x11
f4
x8
f6
x10
x12
Tilt piston
x6
Bucket-Pile interactions
u1-2 : Joysticks action lift/tilt
u3 : Throttle pedal action
x1 : Engine RPM
x2 : Torque to output shaft
x3-4 : Control values lift/tilt
x5-6 : Resistive force lift/tilt
x7-8 : Net force piston lift/tilt
x9-10 : Velocity piston lift/tilt
x11-12 : Position piston lift/tilt

Fig. 2. Propagation of operator control actions to movement of lift and tilt pistons during bucket filling with a wheel-loader. The throttle action (u_3) produces a change in RPM (x_1) and drives the machine via torque on the output shaft (x_2). The RPM changes the pressure in the hydraulic system which together with the lift/tilt action (u_{1-2}) affects the control values (x_{3-4}). The changing fluid flow and interaction forces (x_{5-6}) determine the net forces (x_{7-8}) on the lift/tilt pistons, which results in the motion of the pistons (x_{9-12}). The measured signals are shown in blue color. The functions (f_{2-4}) have varying time-delayed dynamics. Furthermore, the interaction forces (x_{5-6}) are difficult to measure or predict. This motivates the use of machine learning algorithms when modeling this system.

investigation and comparison of six different types of models. We argue that an end–to–end machine learning algorithm addressing this problem can be beneficial in terms of flexibility and efficiency.

II. THE EXPERIMENTAL SETUP

The setup consists of a Volvo 180H wheel loader equipped with additional sensors needed to record the pressures in the lift and tilt hydraulic cylinders. The machine is also modified to read and write signals on the Canbus which are connected to the machine ECUs (engine control units). This gives the possibility to record internal signals like the engine RPM, and the position and velocity of the lift and tilt joints.

The operator determine the actions (lift/tilt joystick and throttle pedal movements) from the sensory input. Fig. 2 illustrates how these actions propagate in the machine to move the bucket through the pile. The bucket-pile interactions are stochastic and difficult to model accurately. Thus, here we assume that conventional modeling from first principles and designing a conventional controller is impractical. Instead, we take a machine-learning approach and train three prediction models, one for each action as shown in Fig. 3.

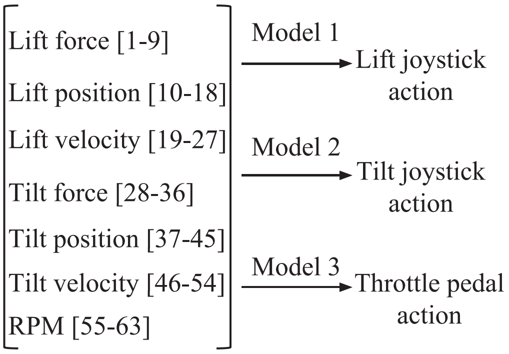

Fig. 3. Input and output variables considered when predicting bucket-filling control actions with the models investigated. The input variables are windowed over eight time steps (400 ms) which results in a total of 63 input signals.

The data used for training is recorded from 150 bucket fillings with one operator loading medium coarse gravel. Some variables (such as the drive shaft speed and gear) appear to have little or no impact on the cross-validation error and are thus excluded here. The training data includes seven variables, as shown in Fig. 3.

There are significant delays in the hydraulics and electronics, and also a small delay from the operator's sensors (vision, tactile) to the actions. To account for the delays, a windowing operator is applied to each input variable so that past values are included ($d = 8$ time steps of 50 ms each). Therefore, each training sample includes $n = 7$ times $(d+1)$ attributes, which in total corresponds to 63 inputs. All inputs and outputs are range-normalized for visualization purposes.

There are three phases of a bucket fill, which we refer to as the entry phase, the scooping phase and the exit phase. The entry and exit phase are straightforward to replicate. In the entry phase, the operator drives the bucket into the pile using the throttle alone. The exit phase is characterized by the use of tilt action alone which eventually results in breakout from the pile. The difficult part is the second phase, where the operator uses the lift joystick, tilt joystick and throttle in combination with each other to fill the bucket. Therefore, we focus on the scooping phase, which lasts for an average duration of 4.76 ($\sigma = 0.7$) sec. The data used to train and test the models correspond to the scooping phase. The data is logged at 20 Hz, which gives an average time-series length of 95 ($\sigma = 15$) for the scooping phase.

III. MODELS

A model that predicts the control actions by the operator needs to account for the complex dynamics of the pile, machine components, human decision making and the operator's body responses. The relationships between inputs and outputs are expected to be non-linear. Thus, we focus on high-capacity models such as neural-networks and regression trees. For comparison and completeness, we also investigate linear regression and k-nearest neighbors and find that these simple models result in higher RMSE.

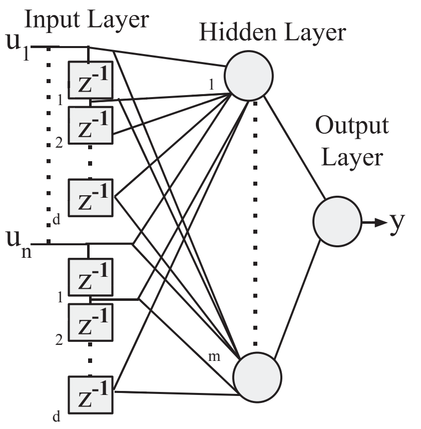

Fig. 4. A feed-forward neural network used for prediction of operator actions with n inputs, d delay units and m hidden neurons.

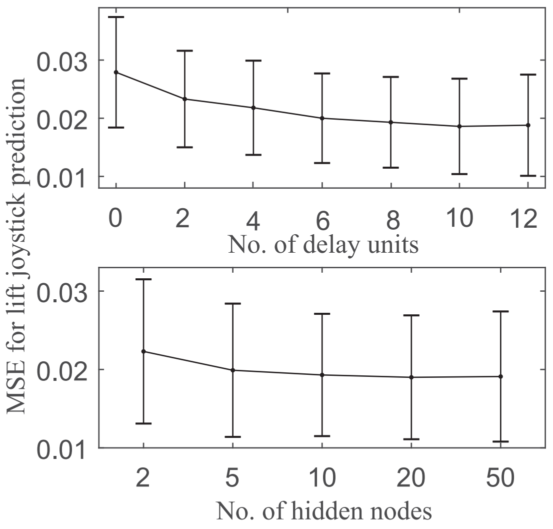

Fig. 5. The preferred number of delay units ($d \sim 8$) and neurons in the hidden layer ($m \sim 10$) are determined by cross-validation.

A. Neural network

Due to the limited volume of training data we consider a small feed-forward neural network. In Fig. 4, we show one example of such a network, for which the preferred number of delay units ($d = 8$) and number of neurons in the hidden layer ($m = 10$) are determined by cross-validation as shown in Fig. 5. The network is trained to minimize the regularized mean square error defined by

$$mse_R = (1-\lambda)\frac{1}{N}\sum_{i=1}^{N}(y_i^p - y_i) + \lambda\frac{1}{n}\sum_{j=1}^{n} w_j^2. \qquad (1)$$

The regularization coefficient, $\lambda = 0.01$, is also determined by cross-validation. The effect of regularization is weak which suggests that over-fitting is not an issue, probably due to the small size of the network.

The hidden-layer neurons implement a *tansig* function

$$a_j = \frac{2}{1+e^{-2\sum_{i=1}^{n(d+1)}(b_i + x_i w_i)}} - 1, \qquad (2)$$

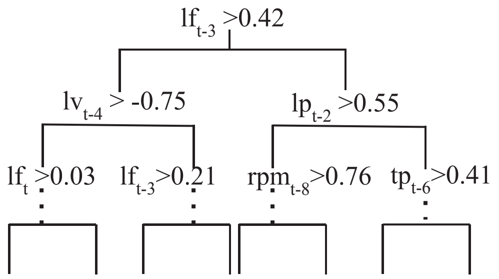

Fig. 6. Trained regression tree for prediction of lift joystick actions. The abbreviations lf, lv, lp, tp denote Lift force, Lift velocity, Lift position and Tilt position, respectively. The lift force is the most important variable for prediction of lift joystick actions.

while the output neurons implement a *poslin* function

$$y = \begin{cases} \sum_{j=1}^{m}(b_j + a_j w_j) & if \sum_{j=1}^{m}(b_j + a_j w_j) > 0, \\ 0 & otherwise. \end{cases} \qquad (3)$$

The networks are trained with resilient back-propagation (*rprop*) algorithm due to its advantages like faster convergence and insensitivity towards parameter selection over standard backpropagation [14]. Faster convergence result in shorter training time, which is beneficial when studying the ensemble models with k-fold cross-validation.

B. Regression trees

We train binary least-squares regression trees using error tolerance as stopping criteria. This implies that the regression tree grows deeper for low values of the error tolerance. An example of a regression tree that is trained to predict the lift joystick actions is illustrated in Fig. 6, which shows that the lift force is the most important parameter for the prediction of lift joystick actions. If the trees are allowed to grow too big (which implies a high number of model parameters), the test performance becomes poor due to overfitting. However, even for trees with optimized size, individual trees show highly jagged responses when compared to the operator's movement of the joysticks. The jagged responses arise from the regression tree algorithm, which is an extension of decision trees that partitions the action space into discrete sections. Thus, to improve the accuracy of the tree-based models, we consider ensembles of regression trees and, for comparison, also ensembles of neural-networks.

C. Ensemble of models

A combination of predictions from many independently trained models can be used to reduce variance of the output and compensate for overfitting. This is the main idea behind ensemble methods like bagging, boosting and random forests. Here, we apply boosting for regression trees and bagging for both neural networks and regression trees. In both cases, the prediction from the individual models are combined by taking the mean of the predictions. Other ways of combining the output from multiple models include optimal linear combination [15], feature-weighted linear stacking [16] and Bayesian model averaging [17].

1) Bagging: We bag several models that are trained on multiple derivations of the training data generated by bootstrapping (random selection with replacement). Bootstrapping is used to obtain a better estimate of the prediction error (and other statistical properties) from a limited sample of data. Since our data are time-series with many separate instances of bucket fillings, we use a simple block bootstrap method. Moving block bootstrap [18] is an interesting alternative approach, which is not investigated here. The number of models in the ensemble of neural networks and regression trees is chosen to 10 and 40, respectively, based on our cross-validation results.

2) Boosting: Boosting differs from bagging in the way that the different models are trained and combined to a resulting model. The models are trained sequentially by reweighting the training samples with the errors such that subsequent models focus more on the samples that previous models did not predict or classify correctly. Each individual model is a weak learner and the combination of all trained models is a strong learner, which have higher accuracy than the individual weak learners.

Bagging aims to reduce variance, while boosting aims to reduce both bias and variance [19]. Boosting is prone to overfitting and typically requires regularization [20]. The assumption of weak learners and the problem of overfitting is handled by limiting the depth of the first tree and the total number of trees in the ensemble.

IV. RESULTS

The cross-validation errors of the predicted control actions by the operator show that the prediction of lift joystick actions is the most difficult problem. Thus, we focus mainly on lift joystick prediction and compare the results obtained with the different models.

The results are obtained with k-fold cross-validation, with $k = 10$. This means that the models are trained with 135 bucket fillings and are validated with 15 bucket fillings in each fold. The validation criteria is the root mean square error (RMSE) defined by

$$RMSE_n = \sqrt{\frac{1}{n}\sum_{i=1}^{len(n)} (y_i^p - y_i)}. \qquad (4)$$

The RMSE of the predicted lift joystick actions is calculated for each individual bucket filling. The best model results in an RMSE less than 0.2 for 148 out of the 150 bucket fillings.

A. Comparison of different models

The mean-RMSE and its standard deviation for all ($N = 150$) bucket fillings are shown in Table I. We find that a small neural network (∼ 650 parameters) can perform better than bagging and boosting ensembles of regression trees with ∼ 67k and ∼ 84k parameters, respectively. When bagging is combined with the neural-network model both the mean and standard deviation of the RMSE is further reduced. The distribution of the RMSE for the six different types of models are shown in Fig. 7.

TABLE I

MEAN AND STANDARD DEVIATION OF RMSE WHEN PREDICTING THE LIFT ACTIONS WITH THE SIX DIFFERENT MODELS.

	μ_{RMSE}	σ_{RMSE}
Mean Model	0.2691	0.0418
Linear Regression (LR)	0.1558	0.0319
k Nearest Neighbors (kNN)	0.1479	0.0315
Bagging RegTree (T-bag)	0.1405	0.0286
Boosting RegTree (T-boost)	0.1390	0.0279
Neural network (NN)	0.1368	0.0278
Bagging NN (NN-bag)	0.1318	0.0270

B. Neural networks and bagging

The RMSE is improved and the variance is reduced when an ensemble of neural networks generated with bagging is used. Fig. 8 shows how an ensemble of ten neural networks predicts the lift joystick action. Individual networks in the ensemble produce similar predictions (shaded region) when the the joystick is moved. However, the shaded region is broader (the predictions by individual networks varies more) when the operator does not move the joystick.

The prediction by the network ensemble is not a smooth function, as can be seen in Fig. 8 (left). If necessary this can be improved with post-filtering and/or a bigger ensemble. In Fig. 8 (middle and right) it appears that the prediction trails the operator action. This can be a sign of an underlining problem arising from the delay in the hydraulics. In the next section, we discuss this aspect and how the lag in the action prediction may be dealt with.

Fig. 9 shows how a single neural network with ten neurons in the hidden layer predicts the different outputs. It can be observed that RMSE of tilt and throttle prediction are lower than of the lift prediction, which is typically the case. This is because the lift joystick is used more by the operator,

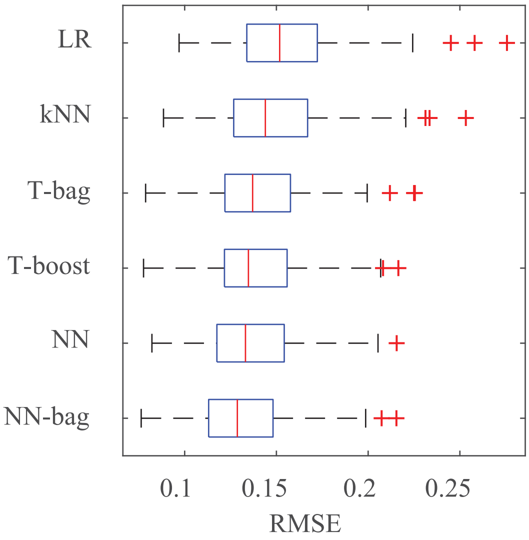

Fig. 7. The distribution of RMSE when predicting the lift actions with each of the six models. A neural network with a single hidden layer outperforms the other model types, and the RMSE is further improved by bagging such networks.

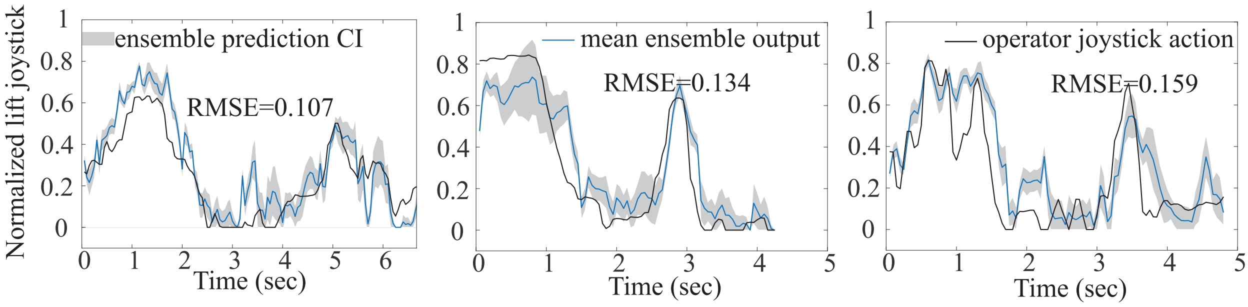

Fig. 8. Prediction of lift joystick actions by a neural network ensemble of ten networks with low (left), average (middle) and high (right) RMSE. The confidence interval (CI) of the ensemble prediction is low when the operator does not move the joystick. The networks in the ensemble agree to a greater extent when the joystick is moved by the operator.

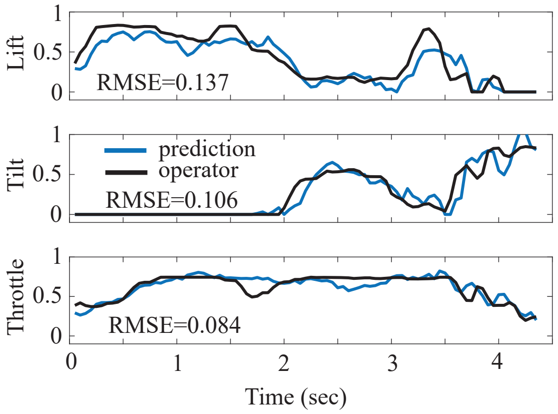

Fig. 9. Prediction of lift joystick, tilt joystick and throttle pedal actions by neural networks with ten hidden neurons. The network hyperparameters are optimized for lift joystick prediction and the same hyperparameters are used when training the networks that predict the tilt joystick and throttle pedal actions.

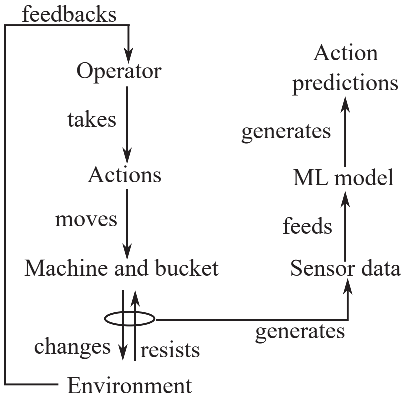

Fig. 10. Flowchart of the bucket-filling problem. The delay in the movement of the machine and bucket after each action by the operator (pedal/joysticks) translates into a delay in the predictions by the machine-learning model.

and also because it fluctuates with higher amplitude. The tilt function takes precedence in the hydraulic system in the sense that the lift piston doesn't move when the tilt joystick is used. Thus, operators learn to use the lift and tilt joysticks alternatively. The throttle action is easiest to predict because the operator strives to maintain a constant RPM when filling the bucket.

V. DISCUSSION

The bucket filling automation problem is addressed with neural networks also in [21], where neural networks are used in a large multi-modular model proposed to automate bucket filling. The main difference compared to this work is that we propose an end-to-end machine learning model for prediction of the actions of an expert operator. The high complexity of the interaction between the bucket and the pile is our motivation for taking a machine learning approach to this problem. A model based approach is considered too inflexible since a change in the excavating conditions can potentially render a proposed solution ineffective. With the machine learning solution, a new model can be trained for each new condition that the machine is used in, and in principle it is possible to further improve the model. For example, a new model can be trained for a different material (some variety of gravel), environmental conditions such as wetness and temperature, and also for a different machine (with similar function).

As it is noticed in Fig. 8 and 9, there is a delay in the model predictions compared to the actions by the operator. This is due to the fact that the models uses the inputs from sensors, which are affected by the actions of the operator after the delay in the control loop. In Fig. 10, the flowchart shows the control loop including the human operator which acts as a controller, and also where the inputs to the prediction model are extracted. Some of the information used as input to the model is thus related to the actions by the operator and some information also originates from the complex interaction between the pile and the bucket.

If an automatic bucket-filling algorithm can be implemented that only requires sensor data as input, the operator loop in Fig. 10 can be omitted and the operator can be replaced by the algorithm. The models presented in this paper can then generate new training data, which can be used to further improve the models to optimize the overall equipment efficiency.

The models presented in this paper are trained to mimic the actions by the operator, which is not necessarily the best strategy for automated bucket filling. The aim of the automatic bucket filling function is to load a given amount of material in minimum time and with minimum fuel consumption and wear on the machine. An open question is whether the models investigated here, which predicts the actions by the operator, can be used to fill the bucket without the assistance from the operator, and how efficient such a solution can be?

The models investigated here can be further improved with a reward function using reinforcement learning [22]. The models are then adapted to maximize the expected future reward by producing behavior associated with a positive reward (such as higher bucket weight and lower bucket-fill time for this problem) and avoiding undesired behavior (like getting stuck, wheel-slip and higher fuel consumption). Reinforcement learning is typically implemented in a Markov-decision-problem (MDP) framework. However, the concept of reward can be used also to address the problem considered here. It can, for example, be used to classify and segregate the training examples with respect to a particular reward. The reward information for each bucket-fill can also be used for modulation of the training protocol.

VI. CONCLUSION AND FUTURE WORK

In this paper, machine learning models are trained on bucket filling data from one type of pile - medium course gravel. Already with 135 examples (~10.7 mins) of training data, the models predict the actions of an experienced operator within an RMSE of 0.2. We train neural networks, regression trees and use ensemble methods to predict the operator control actions. We find that neural networks are more suitable for prediction of these continuous signals than regression trees. A small neural network with ten units in the hidden layer outperforms an ensemble of regression trees.

Currently, the supervised models presented here are implemented in the wheel-loader and tests are performed to evaluate if these models can be used as an automatic controller. The preliminary results demonstrate that this is indeed possible. Thus, an experiment will be designed where the productivity and fuel efficiency of the machine learning bucket filling model and the operator can be compared. If the tests are completed successfully, it will open up the scope for further improvements of neural networks for automatic bucket filling using reinforcement learning (RL). In this work, the start and stop of the the scooping phase is determined by hard-coded rules and parameters. With access to more data, in an online RL setup, such parameters can be further improved. The continuation of the work presented here will focus also on porting the off-line training protocol used here to online learning, including reinforcement learning to enable optimization of a given performance function.

ACKNOWLEDGMENT

The work presented in this article is supported by VINNOVA under the contract 2014-01882. Fredrik Sandin's contribution is supported by the Kempe Foundations under contract Gunnar Öquist Fellow 2014. We thank Erik Uhlin, Martinsson Torbjörn and Benny Johansson for their contributions towards making this study possible.

REFERENCES

[1] S. Dadhich, U. Bodin, and U. Andersson, "Key challenges in automation of earth-moving machines," *Automation in Construction*, vol. 68, pp. 212–222, 2016.

[2] A. Hemami, "Fundamental Analysis of Automatic Excavation," *Journal of Aerospace Engineering*, vol. 8, no. 4, pp. 175–179, 1995.

[3] S. Xiaobo, P. Lever, and W. Fei-Yue, "Fuzzy Behavior Integration and Action Fusion for Robotic Excavation," *IEEE Transactions on Industrial Electronics*, vol. 43, no. 3, pp. 395–402, 1996.

[4] S. Dadhich, U. Bodin, F. Sandin, and U. Andersson, "Machine learning approach to automatic bucket loading," in *2016 24th Mediterranean Conference on Control and Automation (MED)*, June 2016, pp. 1260–1265.

[5] N. Hasan, N. C. Nath, and R. I. Rasel, "A support vector regression model for forecasting rainfall," in *2015 2nd International Conference on Electrical Information and Communication Technologies (EICT)*, Dec 2015, pp. 554–559.

[6] X. Hu, S. E. Li, and Y. Yang, "Advanced machine learning approach for lithium-ion battery state estimation in electric vehicles," *IEEE Transactions on Transportation Electrification*, vol. 2, no. 2, pp. 140–149, June 2016.

[7] R. M. Yedra, F. R. Díaz, M. del Mar Castilla Nieto, and M. R. Arahal, *A Neural Network Model for Energy Consumption Prediction of CIESOL Bioclimatic Building*. Cham: Springer International Publishing, 2014, pp. 51–60.

[8] H. R. Khosravani, M. D. M. Castilla, M. Berenguel, A. E. Ruano, and P. M. Ferreira, "A comparison of energy consumption prediction models based on neural networks of a bioclimatic building," *Energies*, vol. 9, no. 1, 2016. [Online]. Available: http://www.mdpi.com/1996-1073/9/1/57

[9] K. Gajowniczek and T. Ząbkowski, "Electricity forecasting on the individual household level enhanced based on activity patterns," *PLOS ONE*, vol. 12, no. 4, pp. 1–26, 04 2017. [Online]. Available: https://doi.org/10.1371/journal.pone.0174098

[10] A. E. Ruano and P. M. Ferreira, "Neural network based hvac predictive control," *IFAC Proceedings Volumes*, vol. 47, no. 3, pp. 3617–3622, 2014, 19th IFAC World Congress. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S147466701642166X

[11] L. Villani and J. De Schutter, *Force Control*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 161–185.

[12] S. Sarata, H. Osumi, Y. Kawai, and F. Tomita, "Trajectory arrangement based on resistance force and shape of pile at scooping motion," in *International Conference on Robotics and Automation*, vol. 4, April 2004, pp. 3488–3493.

[13] A. A. Dobson, J. A. Marshall, and J. Larsson, "Admittance control for robotic loading: Design and experiments with a 1-tonne loader and a 14-tonne load-haul-dump machine," *Journal of Field Robotics*, vol. 34, no. 1, pp. 123–150, 2017. [Online]. Available: http://dx.doi.org/10.1002/rob.21654

[14] M. Riedmiller and I. Rprop, "Rprop - description and implementation details," 1994.

[15] S. Hashem, "Optimal linear combinations of neural networks," *Neural Networks*, vol. 10, no. 4, pp. 599 – 614, 1997. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0893608096000986

[16] J. Sill, G. Takács, L. W. Mackey, and D. Lin, "Feature-weighted linear stacking," *CoRR*, vol. abs/0911.0460, 2009.

[17] P. Domingos, "Bayesian averaging of classifiers and the overfitting problem," in *IN PROC. 17TH INTERNATIONAL CONF. ON MACHINE LEARNING*. Morgan Kaufmann, 2000, pp. 223–230.

[18] H. R. Kunsch, "The jackknife and the bootstrap for general stationary observations," *The Annals of Statistics*, vol. 17, no. 3, pp. 1217–1241, 1989. [Online]. Available: http://www.jstor.org/stable/2241719

[19] T. G. Dietterich, *Ensemble Methods in Machine Learning*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2000, pp. 1–15.

[20] W. Jiang, "On weak base hypotheses and their implications for boosting regression and classification," *Annals of Statistics*, vol. 30, no. 1, pp. 51–73, 2002.

[21] L. Wu, "A study on automatic control of wheel loaders in rock/soil loading," Ph.D. dissertation, University of Arizona, 2003.

[22] R. S. Sutton and A. G. Barto, *Introduction to Reinforcement Learning*, 1st ed. Cambridge, MA, USA: MIT Press, 1998.