Automation in Construction 113 (2020) 103138

Contents lists available at ScienceDirect

Automation in Construction

journal homepage: www.elsevier.com/locate/autcon

Construction activity recognition with convolutional recurrent networks

Trevor Slaton[a], Carlos Hernandez[b], Reza Akhavian[c,*]

[a] Department of Mathematics, California State University East Bay, 25800 Carlos Bee Blvd., Hayward, CA 94542, United States of America
[b] School of Engineering, California State University East Bay, 25800 Carlos Bee Blvd., Hayward, CA 94542, United States of America
[c] Department of Civil, Construction, and Environmental Engineering, San Diego State University, San Diego, CA 92182, United States of America

A B S T R A C T

Although heavy equipment is an indispensable resource in many construction projects, it is often underutilized. Inefficient usage patterns and frequent idling contribute to increased emissions and project costs. Efforts to improve usage patterns often begin with activity tracking. Recent research into automated activity tracking has leveraged sensing devices and Internet-of-Things (IoT) frameworks to power machine learning models that can predict the behaviors of monitored equipment. However, shallow machine learning models require complex manual feature engineering that could be further automated with more recent deep learning approaches. Deep learning approaches not only increase automation but also promise improved accuracies by avoiding biases introduced by manual feature design. This paper proposes a construction equipment activity recognition framework that uses deep learning architectures to predict the activities of heavy construction equipment monitored via accelerometers and applies this framework to a roller compactor and an excavator performing real work. The performance of a simple baseline convolutional neural network (CNN) is compared to a hybrid network that contains both convolutional and recurrent long short-term memory (LSTM) layers. The hybrid model outperforms the baseline model in all instances studied. In the task of classifying the activities of the roller compactor, the hybrid model achieves a validation accuracy of 77.1% when presented with six activities and a validation accuracy of 96.2% when distinguishing only direction. In the task of classifying seven activities of the excavator, the hybrid model achieves a validation accuracy of 77.6%, with some confusion between isolated activities and a *Various* category that includes elements of the isolated activities. With the *Various* category removed, the hybrid model achieves a validation accuracy of 90.7%. This study demonstrates that deep learning frameworks can model the activities of construction equipment with high accuracy. In particular, this work shows that convolutional and LSTM layers can each form effective parts of deep learning models that characterize equipment activities based on accelerometer data, and furthermore that these components can produce more effective models when combined. The findings of this study can be leveraged by researchers and industry professionals to develop reliable automated activity recognition systems for tracking and monitoring equipment performance and for measuring the productivity and the efficiency of the work performed.

1. Introduction

The architecture, engineering, and construction (AEC) industry is increasingly investing in disruptive technologies. Technological advancements combined with data analytics enable enhanced design, planning, and construction [1]. These tools can be leveraged to overcome longstanding AEC challenges such as schedule conflicts, budget overruns, and risk management. Effective applications of the right technologies can produce measurable gains in productivity, sustainability, and safety over the course of a project. Automated tracking and monitoring of resources is a good example of one application of technology that enables these improvements.

According to industry reports, approximately 35% of the total working time across the entire construction workforce (i.e. 14 h/week on average, including office and field workers) is consumed performing non-productive tasks such as searching project information, solving conflicts, waiting for instructions, or wasting time performing poorly planned activities [2]. At a productivity rate of only 43%, the construction industry lags behind other industrial sectors such as manufacturing, which has more than double the productivity at a rate of 88% [3]. It is vital that the construction industry overcomes this problem. Research into systems that leverage new technologies and novel methodologies to increase awareness of performance issues and enhance productivity is well underway. Some researchers focus on the problem at a higher level. For instance, Cheng et al. [4] developed an automated system that monitors the overall progress of a project's execution. Others focus on tracking resource usage at the level of individual workers or at the level of individual pieces of equipment. To that end, Akhavian and Behzadan [5] implemented a motion sensor-based tracking system to study the productivity of individual construction workers performing a variety of tasks. Cheng et al. [6] noted that productivity rate tends to vary across phases of a construction project, an insight that many construction practitioners do not consider very often. Similarly, Wideman [7] discovered that, while productivity rates across different sectors of the construction industry are inconsistent, productivity generally tends to be lowest during the initial

* Corresponding author.
E-mail addresses: tslaton@horizon.csueastbay.edu (T. Slaton), chernandez233@horizon.csueastbay.edu (C. Hernandez), rakhavian@sdsu.edu (R. Akhavian).

https://doi.org/10.1016/j.autcon.2020.103138
Received 19 August 2019; Received in revised form 6 February 2020; Accepted 15 February 2020

phases of any construction project and grows substantially as it nears completion. Heavy construction, however, suffers from poor performance: it showed the lowest productivity growth in comparison with other construction in single-family, multi-family, and industrial areas [8]. The utilization of heavy construction equipment is heavily correlated with the construction industry's productivity problem. In this regard, Gong and Caldas [9] determined that usage of heavy construction equipment is most intense during the initial phases of construction projects. Therefore, giving special attention to heavy construction equipment in productivity analysis is critical for seeking improvements in resource management. Monitoring heavy equipment activities aids construction managers in their efforts to minimize the amount of time that their fleets spend performing non-value-adding activities. Reducing such activities will increase productivity and reduce the environmental impact associated with running such machines.

Automated identification of the activities performed by the different construction resources has been the subject of many recent studies. The overarching goal is often to develop an Internet-of-Things (IoT) framework that uses machine learning techniques to distinguish different activities performed by construction workers and/or construction equipment, based on data collected from various sensors. Human activity recognition (HAR) has been extensively explored in other fields [10] as well as in the construction literature [11]. However, the nature of activities performed by heavy construction equipment, the different degrees-of-freedom of their various articulated parts, and the rugged terrains on which they operate pose additional challenges. The present journal paper develops a state-of-the-art methodology for construction equipment activity recognition by combining the cost-effectiveness, unobtrusiveness, and reliability of wireless inertial measurement units (IMUs) suggested by Akhavian and Behzadan [12] with the high accuracies and unprecedented degrees of automation offered by modern deep learning techniques. This methodology lays the foundation for future work that will extend the models to predict the greenhouse gas (GHG) emissions associated with construction equipment's activities. Since deep learning techniques automate feature extraction, they lead not only to higher classification accuracies but also to models that are simpler to adapt to different kinds of equipment. This topic has been gaining traction in the construction research community very recently which demonstrates the feasibility and effectiveness of the approach [13–15].

The previous studies in this are focused on vision-based activity recognition or evaluation of models trained with synthetic data such as those developed using data augmentation methods. Such studies have established a compelling precedent on the importance of this topic. The presented paper differs from the past studies since it is an early attempt to deploy deep learning using inertial sensor data collected from more than one piece of equipment in an uncontrolled environment. Recent studies in this area are reviewed in detail in the Research Background section where the contribution of the research presented in this paper is highlighted in more detail.

The remainder of this document is organized as follows: in the next section, a comprehensive literature review is presented to outline the latest research findings on the use of sensors for activity recognition and analysis. Next, the research methodology and data collection setup are described. Results are then presented for different types of equipment. Finally, a detailed discussion of the results is provided, conclusions are drawn, and future research directions are outlined.

2. Research background

2.1. Sensing approaches for activity recognition

Traditionally, construction equipment performance is recorded manually by direct observation onsite. This manual method of monitoring could be subject to error and inefficiency [16]. As such, there is a great deal of research focused on automating the monitoring of construction workflows. Most automated construction equipment monitoring approaches rely on one of three broad categories of sensor modalities. The first class of sensing techniques employs motion sensing devices such as single accelerometers [17] or IMUs [12]. The second class of sensing techniques employs cameras to capture video streams, which are processed using computer vision algorithms [18]. Sometimes the cameras are stationary, while other times they move along with the equipment being tracked. A third class of sensing approaches uses microphones to map different tasks to audio signatures produced by machines while they work [6]. The latter two approaches can yield promising results; however, the chaotic nature of construction jobsites tends to disrupt them. Cameras struggle to maintain line of sight, and sound sensors (i.e. microphones) struggle to pick out signal amidst obstructions and background noise. There are some other technologies such as vehicle health monitoring systems (VHMS) embedded in newer construction equipment by the original equipment manufacturer (OEM) that provide accurate performance tracking. However, retrofitting old equipment with these systems can be costly and subject to significant compatibility problems [19,20]. Real-time locating systems (RTLS) based on global positioning system (GPS) and ultra-wideband (UWB) technologies are also options for tracking construction resources. These technologies have been used to monitor everyday activities in natural contexts, as well as the activities of nursing personnel [21,22]. In construction, such technologies have been leveraged to extract queuing properties from heavy construction operations and to assess real-time safety risks at hydropower construction sites [23,24]. Unfortunately, these systems cannot provide direct insight into complex articulated activities carried out at particular locations, as they are only aware of high-level location information.

Given the drawbacks of other sensing approaches, motion sensing techniques are an increasingly popular option. IMUs are inexpensive, easy to obtain, and compact enough that they can be mounted on construction machinery without any complicated procedures or costly modifications. Compared to lone accelerometers, IMUs offer the additional advantage of integrating complementary sensors like gyroscopes and magnetometers that can provide a fuller picture of the monitored activities. IMUs have been applied successfully to human activity recognition tasks such as the Joshua and Varghese [17] study of mason workers' productivities and the Akhavian and Behzadan [5] study of construction workers' activities using smartphone sensors. In other studies, such as the accelerometer-based study of three excavator activities by Ahn et al. [25] and the IMU-based front loader study by Akhavian and Behzadan [12], motion sensing devices were shown to be viable sources of data for construction equipment activity analysis. The methodology developed here builds on this prior work but achieves enhanced results because of the differences detailed in the following sections.

2.2. Machine learning approaches for activity recognition

Deep learning models leverage many cascaded layers of nonlinear information processing to fit patterns in input data with much greater capacity than traditional machine learning models [26]. Human activity recognition frameworks have benefitted enormously from this new technology. Yang et al. [27] showed convolutional neural networks (CNNs) can accurately classify the activities of human subjects. Ordoñez and Roggen (2016) reached a new state of the art for distinguishing complex human activities by developing a long short-term memory (LSTM) network. Both of these models were developed using inertial sensor readings. One of the most compelling advantages that deep learning models offer over their shallower counterparts is the potential for automatic feature extraction. In traditional machine learning models, feature selection is a heuristic process with lots of trial and error and human-guided design. Such processes can cut out features that might be critical before the model ever sees them or present the data in such a way as to confer the biases of their designer to the trained

model [28]. By employing expressive deep learning models that avoid such biases, the work presented here is able to achieve unprecedented accuracies in classifying many complex activities of heavy machinery despite the added challenge of working with machines performing real work in hectic, live construction sites.

2.3. Deep learning approaches for construction equipment activity recognition

Considering the advantages provided by deep learning algorithms versus shallow models, their application in construction activity recognition has been explored very recently by some researchers. In one study by Kim and Chi [14], excavators performing earthmoving operations were subject to vision-based activity recognition. Models containing the Faster Region-proposal Convolutional Neural Network (Faster R-CNN) developed by Ren et al. [29] and Double-layer Long Short-Term Memory (DLSTM) layers were trained on data collected in experimental settings. The methodology resulted 90.9% precision and 89.2% recall rates. Another vision-based research study used LSTM networks to identify construction entities and their activities by focusing on spatial states and attentional cues [13]. These studies offer promising solutions in construction environments where vision-based methods can be applied. The presented research, however, develops models on inertial sensors that can be used in more varied construction environments and extends the methodology to more than one type of equipment.

In another study, Rashid and Louis [15] proposed time-series data augmentation to generate synthetic training data. Their work suggests that time series data augmentation can greatly improve the performance of construction activity recognition models dealing with otherwise limited training data. However, their study focuses on the effects of data augmentation on an LSTM-based model without considering the possible effects of convolutional neural network layers like the present study. Additionally, their work was performed in a controlled environment on a single type of equipment. The work presented here builds upon the findings of an earlier research project by the authors [30] and studies two kinds of equipment performing real work.

3. Data collection

All data collection sessions involved construction equipment performing real work without any special directions. The first data collection session focused on a BOMAG BW 145PDH-3 single drum vibratory roller compactor executing landscaping tasks and working on a driveway for a hotel construction project in San Jose, California. The second data collection session focused on a CAT 328D crawler excavator digging a trench at a sewage treatment plant in Pinole, California. Both sessions were recorded on video using a Logitech C270 webcam; however, the videos were used only for data annotation since the aim of the data collection is to develop activity prediction models dependent only on accelerometer readings once trained. Two MyoMotion 684 accelerometer sensors by Noraxon were mounted on different movable of parts each machine studied [31]. To collect readings in real-time, the sensors communicated with a nearby laptop outfitted with a radio antenna. The sensor kit included software to synchronize the sensor readings with the recorded video, which was critical for labeling the data accurately based on the activities observed in the video. The setup used for data collection is shown in Fig. 1.

In both data collection sessions, the first sensor was placed inside the cabin on the dashboard. In the roller compactor data collection session, the second sensor was secured to the roller's support arm; in the excavator data collection session, the second sensor was attached to the excavator arm near the bucket. Fig. 2 shows the sensors' placements on each piece of equipment studied.

The sensor in the cabin was meant to capture overall movements of the equipment without too much disturbance from the articulations of its arm; the sensor placed on the arm was intended to focus on the articulations of the arm which are of particular relevance to the various activities studied. The different parts of the data collection process including the laptop and the receiver, the sensors attached to the equipment body streaming motion data, and the video recording were all coordinated synchronously for later data labeling and analysis. After being attached to the equipment, the sensors were calibrated to reorient their three axes of motion detection and eliminate considerations of their physical orientations (e.g., possibly upside down).

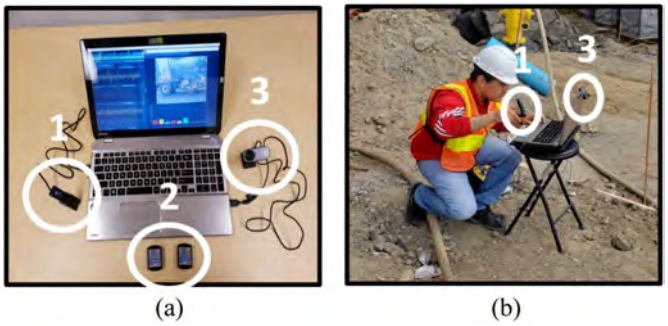

Fig. 1. (a) The Noraxon sensor set: 1. Radio antenna 2. MyoMotion sensors 3. Logitech C270 webcam and (b) Data collection underway at an actual jobsite.

Each sensor provided three channels worth of acceleration data – one for each of the x, y, and z axes – and recorded each channel at 100 Hz (100 readings per second). Using two three-axis accelerometers recording motions in each axis at 100 Hz in each experiment produced six channels of 116,536 sensor readings describing 20 min of compactor activity and six channels of 173,600 sensor readings describing 30 min of excavator activity. These readings were captured in a CVS file for further processing.

The video recording was synchronized to the sensor data via software running on the laptop. The activities performed in the video recording were manually labeled, and these activity labels were added to the CSV file containing the accelerometer data based on time stamps, since those readings where synchronized with the video. The resulting CSV file consisted of all of the accelerometer readings with activity labels at each time step, taken to be the ground truth activity labels. The data in the CSV files were split into training and validation subsets and further processed as detailed in the Data Analysis section.

4. Methodology

Prior work established the state of the art in human gesture recognition using a neural network design operating on data from various sensors called *DeepConvLSTM* [32]. The work presented here builds off this work, adapting the model to the task of predicting construction equipment activities from accelerometer readings. Although intuition may suggest the problems have a lot in common, the movement patterns of heavy machinery are quite different from those of human subjects making typical gestures. Some modifications were necessary, but the major principles of the *DeepConvLSTM* architecture do in fact translate well to the construction activity domain. That is, combining convolutional layers to extract locally correlated features from sensor data with long short-term memory (LSTM) layers to process evolutions of those features across time is an effective paradigm for construction equipment activity recognition. Fig. 3 shows an overview of the approach developed here.

Typically, an LSTM uses fully-connected networks instead of convolutional networks within its internal networks, although there is some research that uses convolutional networks within an LSTM [33]. That work was also shown to benefit from convolutional structures, but the work presented here did not study a model featuring convolutional structures inside LSTMs. Instead the convolutional layers are preceding

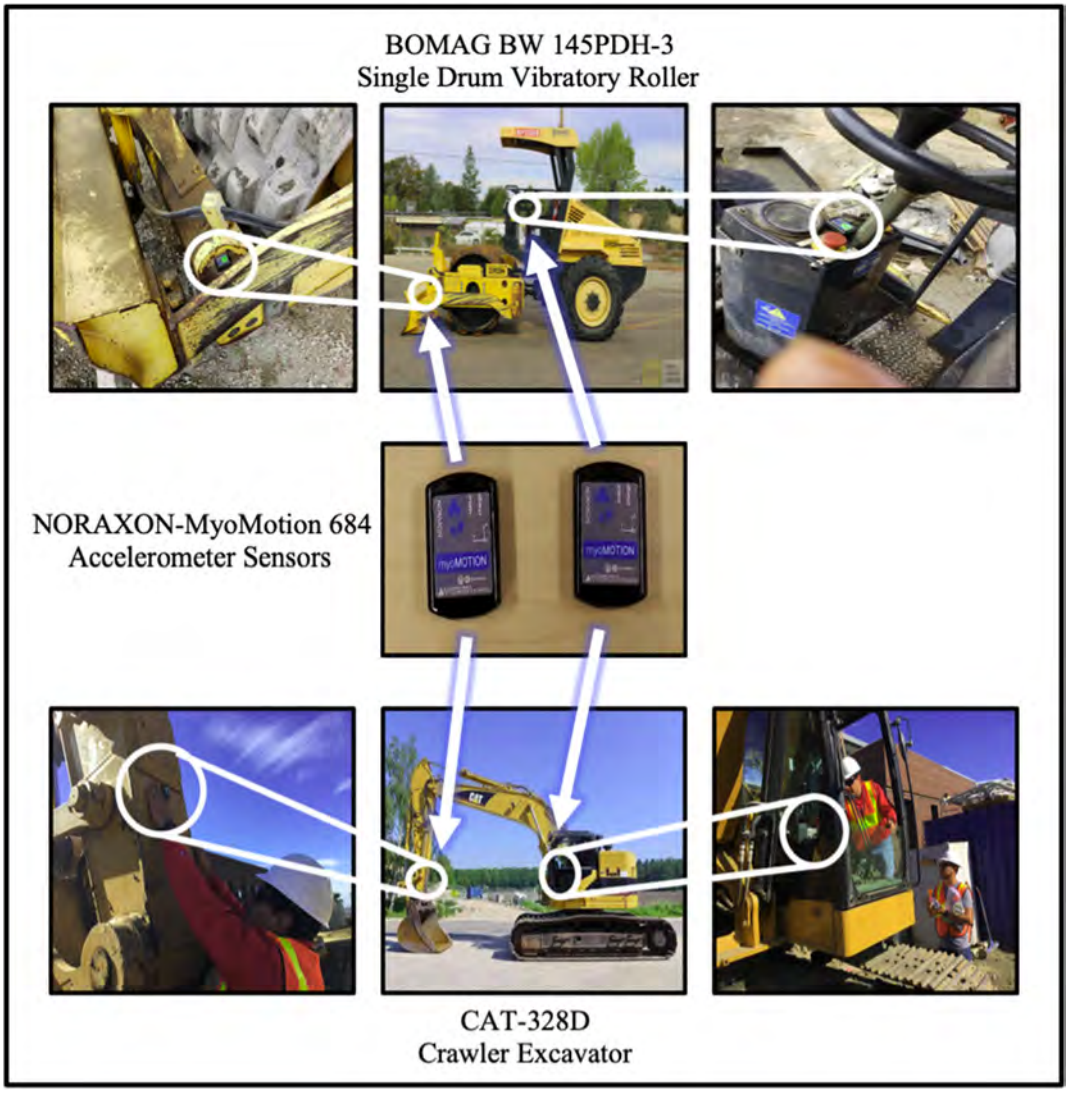

Fig. 2. The locations of the accelerometers installed on the studied machines.

the standard LSTM layers. Fig. 4 illustrates differences between the structures of a fully-connected network and a convolutional network.

Because a fully-connected network has a completely filled-in computation graph, it is theoretically possible that it could learn the same weights as a convolutional network and therefore extract the same features. However, in practice the reduced connections and weight sharing in a convolutional neural network encourage the model to learn patterns related to features with strong local correlations (i.e. samples near each other in time are more likely to contribute to a single feature). Convolution is explained in more detail after the next section.

4.1. Data analysis

Models were trained and validated on disjoint subsets of the data. Validating on data not seen during training provides insight into real-world predictive power. Across the data collected, the final 20–28% of sensor readings were set aside for validation; the rest of the data were used for training. Care was taken so as to split the data from each data collection session into two contiguous time series (i.e. sensor readings that were taken in order without any time gap between them) preserving similar distributions of activity labels across the partitions. In order for the learning task to be feasible, it is critical that the data used for prediction (i.e., the validation sets) have statistical properties similar to the data used to train the models.

For the roller compactor data, the training set consisted of the first 92,728 contiguous samples while the validation set consisted of the remaining samples. Some activity labels (*Idle* and *Off*) were too rare in the collected data to appear in both the training and validation sets. As these activities occurred only during the first 1040 samples and during the last 8017 samples of the data, it was easy to drop them without disrupting the time series. Fig. 5 plots the activities the roller compactor performed during the data collection session. In the full problem, all six activity classes shown in the white and blue regions were considered; however, separate models were also trained with different combinations of activities combined into a single category to study the simplified subproblems of dealing only with the machine's movement directions and dealing only with the machine's vibration settings in isolation.

It was then evaluated on a subset of the validation data excluding all frames labeled *Various*, as well as on a subset additionally excluding the first 14,335 validation frames labeled *Idle* to rebalance the class distribution, which shifted significantly upon removal of the *Various* frames. The model was able to identify the *Idle* activity with nearly perfect accuracy, so the rebalanced scenario posed a more realistic challenge.

For the excavator, the first 125,165 contiguous samples were used for training, and the remaining 48,435 samples were used for validation. Fig. 6 depicts the data used for the excavator experiment. Transitions between activities in this dataset were much more frequent than

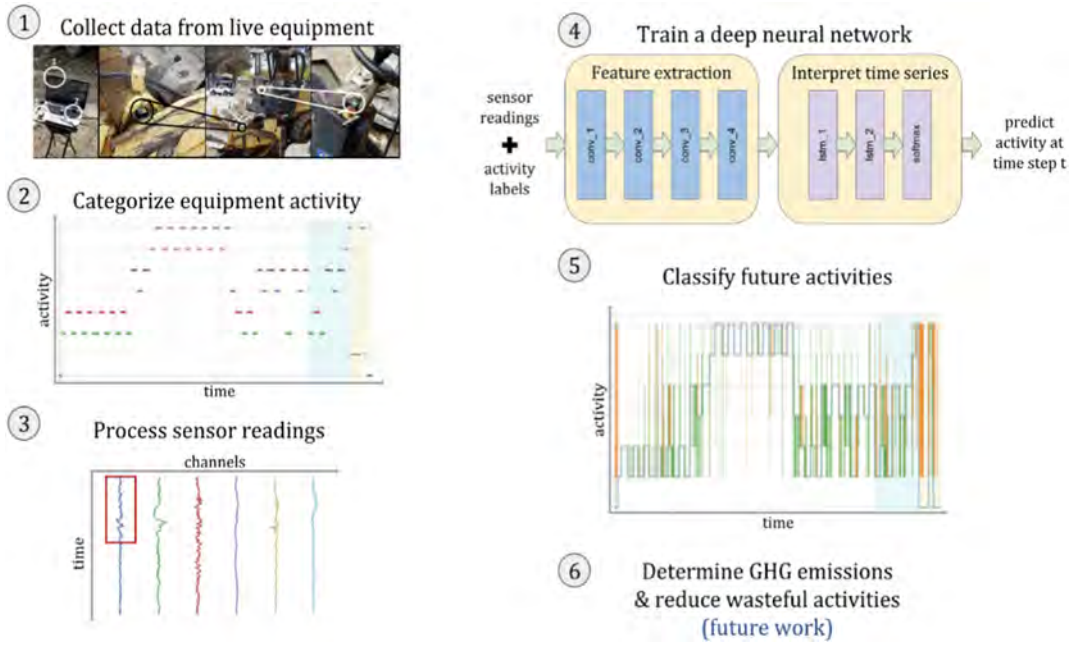

Fig. 3. An overview of the developed methodology.

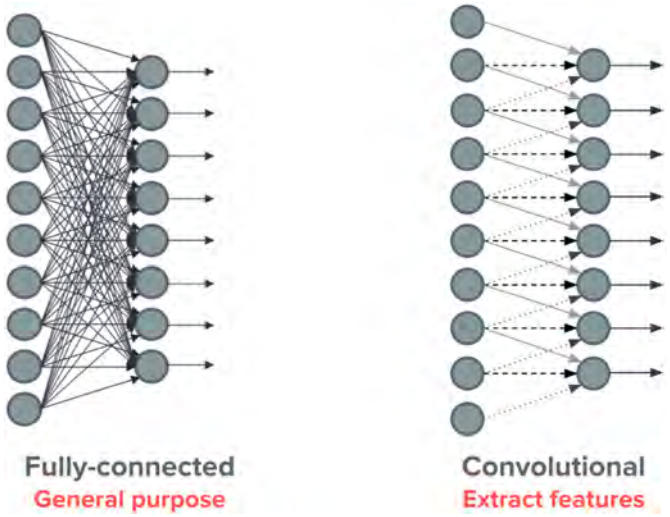

Fig. 4. The computation graphs connecting two layers of a fully-connected network and two layers of a convolutional network are compared above. The convolutional network drops many connections and reuses weights depicted by arrows with the same visual style.

in the roller compactor dataset. No samples were dropped; however, in order to consider the subproblem of identifying only the excavator's isolated activities without the confusion caused by the *Various* activity category, the model was additionally trained from scratch on the subset of the training frames with activity labels other than *Various* (refer to the paragraph after the next one to see how frames are computed).

It is worth mentioning here that the choice of the activity classes selected for classification and prediction in this study is based on two main considerations. First, activities with a high level of detail are more difficult to distinguish by machine learning models. Therefore, a rather high level of detail (LoD) is considered here as a technical complexity to evaluate the performance of the developed algorithm. If the model is successful in detecting and distinguishing among fine-grained activity labels such as *Rotating (Left)* and *Rotating (Right)* or *Forward (high)* and *Forward (low)*, for example, one can use the attributes of these activities to get insight into the characteristics of coarser-grained activities. For instance, if activities a and b are performed in a row, a new activity c of a coarser-grain and comprising of a and b has a duration of $t_c = t_a + t_b$. A comprehensive discussion on the LoDs is presented in a previous publication by the third author [12]. The second consideration for these activity classes, especially in the case of the roller, is the equipment emission and fuel consumption level which have different values in these different types of activities [34]. The outcome of this project can inform future research on activity recognition-based emission estimation based on distinct engine tiers and power modes.

A single set of accelerometer readings at a given time step would be meaningless without being placed in a larger context of recent accelerometer readings. To capture the temporal contexts critical to the problem, the six channels of sensor readings were segmented into overlapping frames of 2 s worth of activity each. Since the sensors recorded data at a sampling rate of 100 Hz, 2 s of activity corresponds to 200 samples. The frames were computed by running a sliding window 200 samples wide across the raw data. Each frame was labeled according to the activity at the last sample in the frame, and the sliding window advanced 1 sample at a time so that 199 samples worth of context were provided for every time step. Thus, the models were tasked with predicting the activity label at every time step, given 2 s worth of context. Larger windows might be chosen at the cost of greater computational complexity while smaller windows might be useful in real-time monitoring applications where a 2 s lead time is unacceptable. For this problem, 2 s frames were appropriate.

The training and validation sets were segmented into frames independently to prevent validation data from leaking into the training set at the boundary between the data set partitions. Before feeding the segmented frames into the models, they were oriented so as to place time on the vertical axis, with the six sensor channels running side-by-side horizontally. Each of the sensor channels was normalized to fall in the range [0, 1]. Fig. 7 depicts an example data frame.

The red box represents a (3, 1) filter that the models slide across each channel in the frame while computing convolutional features. It is not drawn to scale. In reality, its height would be 3 cs (0.03 s), and its width would be 1 channel. The signals are labeled by the accelerometer axis they represent (x, y, z) and subscripted with the number of the sensor they belong to.

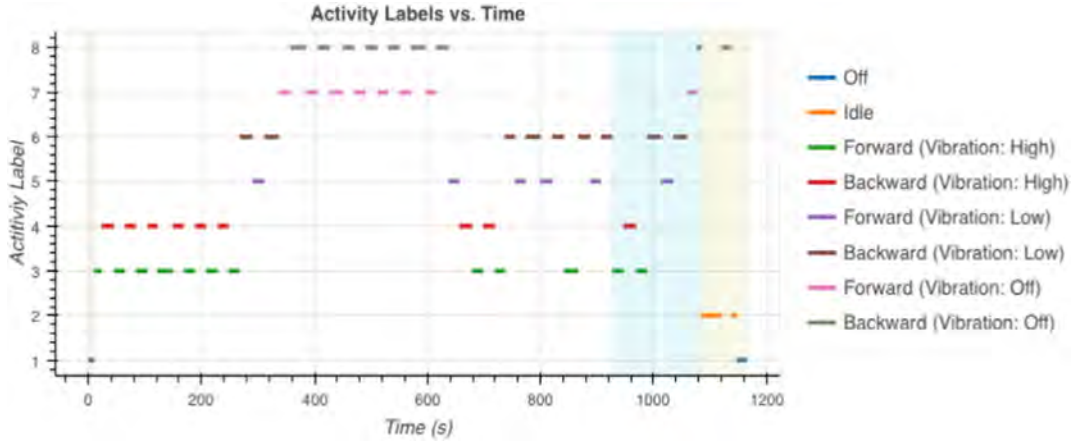

Fig. 5. Activity data vs. time for the roller compactor experiments. The data used for training lies in the white region, the validation data in the blue, and data not considered in the yellow. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

4.2. Convolution

Eq. (1) describes how to convolve a 1D filter K of width W with discrete input signal $X(\tau)$:

$$(K * X)(\tau) = \sum_{i=1}^{W} K(i)X(\tau - i) \tag{1}$$

To convolve a 2D filter K of width W and height H with $X(\tau)$,Eq. (2) can be applied:

$$(K * X)(\tau) = \sum_{i=1}^{H}\sum_{j=1}^{W} K(i,j)X(\tau - i, \tau - j) \tag{2}$$

The pattern extends to arbitrary dimensionality. An intuitive take on these sorts of computations is that the filter slides over the input signal in its various dimensions and computes a new signal. As the filter slides, the (i,j) component of the new signal is the result of centering the filter at (i,j) in the input signal and computing the sum of the element-wise products between the filter's entries and the corresponding values of the input signal, wherever they overlap.

Convolutional filters are often used as feature detectors. Convolutional neural networks typically make use of multiple filters in each layer, which allows them to compute multiple features interpreting different aspects of the data they are processing. Deep convolutional networks can then build higher-level, more abstract feature detectors by applying the filters in later layers to the features detected by earlier layers.

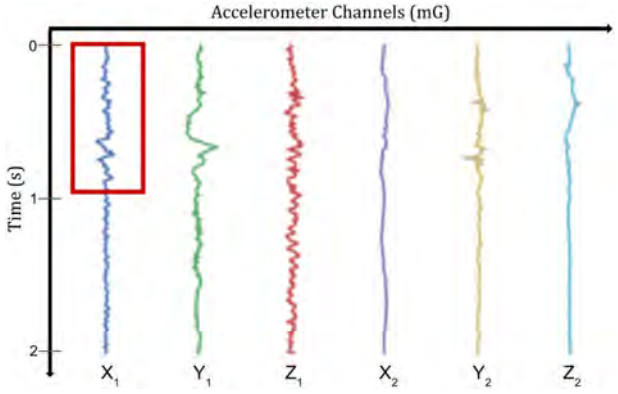

Fig. 7. A 2-second frame of sensor data computed by the sliding window preprocessing step.

4.3. Deep learning models

The Python library Keras was used to implement all of the models studied on top of a TensorFlow backend. Models studied include *BaselineCNN*, a fairly standard convolutional neural network, and *DeepConvLSTM*, a more sophisticated network that combines convolutional layers with recurrent LSTM layers. Although these models were heavily inspired by the work of Ordóñez and Roggen [32] in human activity recognition, a few significant changes were made. First, to

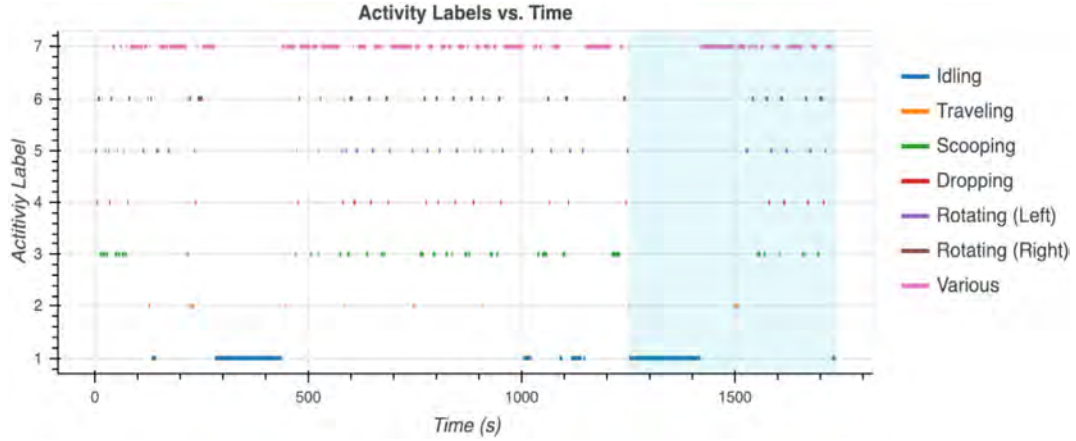

Fig. 6. Activity data vs. time for the excavator experiment. Validation set in blue region. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

speed up convergence during training, batch normalization layers were inserted between each convolutional layer. Also, a dropout layer was added between the block of convolutional feature extractors and the rest of the network. This layer randomly deactivates 25% of the features passed between these segments of the network per batch during training as an additional form of regularization. In other words, it discourages learning overdependence on any particular feature to improve the model's generalization abilities [35]. The convolutional baseline and hybrid convolutional-recurrent architectures intentionally share many characteristics. Both start with a block of four consecutive convolutional layers, each of which learns 64 different filters of size (3, 1). These filters apply convolutions only along the time axis of the input frames to ensure that features derived from each sensor channel remain independent until they reach later layers of the network. Max pooling is omitted to better preserve the translational equivariance of the convolutional feature extractors because the positions of features along the time axis within the frames are of critical importance for this problem. The two models studied diverge in terms of the architectures they use to interpret the outputs of this common feature extracting block. *BaselineCNN* uses two fully-connected layers with 128 neurons each, followed by a softmax classifier, to compute a likelihood score for each activity label. The largest score is interpreted as the model's activity prediction. *DeepConvLSTM* uses LSTM layers instead of the fully-connected layers to make its activity predictions. To keep the LSTM layers comparable to the baseline's fully-connected layers, they use state vectors of size 128. Networks inside the LSTM layers learn to manage their state vectors like a memory containing only the most salient details of an observed sequence that is otherwise too large recall [36]. Because LSTMs' memories are better suited to dealing with time series, higher performance is expected from *DeepConvLSTM* compared to *BaselineCNN*. Fig. 8 illustrates two activation functions commonly used in networks inside each LSTM layer — *tanh* and the sigmoid function (σ) — while Fig. 9 illustrates what goes on inside an LSTM layer, alongside its equations.

Perhaps the simplest kind of neural network is the multilayer perceptron (MLP). A perceptron computes a linear output by multiplying an input vector x by a matrix of learned weights W and adding a learned bias term b, just like a vectorized form of the classic linear equation $y = mx + b$.An MLP builds a non-linear (potentially deep) model by feeding a perceptron's output through a non-linear activation function like the sigmoid function σ and cascading several such structures. The generic equation for the output of a single layer of a multilayer perceptron is thus $y = \sigma(W \cdot x + b)$. Building an MLP with all possible connections between adjacent layers reflected in the weights W results in the usual fully-connected network.

An LSTM cell holds its memories in a state vector C, which can have arbitrary size (128 in *DeepConvLSTM*). When provided a sequence input x, the LSTM cell processes each value in the sequence at time t in a recurrent fashion, producing an output h_t for each input x_t. In addition to x_t, the LSTM cell considers its previous state vector $C_{t-1}$and its previous output $h_{t-1}$when calculating its current state C_t and current output h_t. The state C_t is calculated according to Eq. (3):

$$C_t = f_t \bullet C_{t-1} + i_t \bullet \widehat{C_t} \text{ where } f_t = \sigma(W_f \bullet [h_{t-1}, x_t] + b_f) \tag{3}$$

The quantity f_t is often called the "forget gate" because it controls which parts of the previous state C_{t-1} are erased. A similar quantity i_t controls which parts of the tentative state $\widehat{C_t}$ get remembered, according to Eq. (4):

$$i_t = \sigma(W_i \bullet [h_{t-1}, x_t] + b_i) \tag{4}$$

Both f_t and i_t are calculated just as if they were layers of an MLP. The concatenation of the previous output h_{t-1} and current input x_t, $[h_{t-1}, x_t]$, gets multiplied by a linear weight matrix W, adjusted by a bias term b, then molded to fit the range $[0, 1]$ by the sigmoid activation function. Ultimately, f_t and i_t are vectors of values between 0 and 1, so they are naturally viewed as weight vectors determining which parts of the old and candidate states determine the current state C_t after element-wise multiplication. The output at each time step is dependent on the calculated state and is given by Eq. (5):

$$h_t = o_t \bullet tanh\,tanh\,(C_t), \text{ where } o_t = \sigma(W_o \bullet [h_{t-1}, x_t] + b_o) \tag{5}$$

Here o_t is another weight vector with values in $[0, 1]$ calculated by the familiar MLP equation. Note that the output of *tanh* is in the range $[-1, 1]$.

5. Results

5.1. Training

Each model was trained for five epochs of batched gradient descent, running Adam with a batch size of 100 frames and a learning rate of 0.001. Adam is a variation on the standard stochastic gradient descent optimization algorithm that adjusts the learning rate based on a running average and the running variance of the recent gradients, which often speeds up convergence [38]. LSTMs are sometimes difficult to train due to cascading gradients; to avoid this problem, gradient clipping was applied with a maximum gradient value of 0.5 and a maximum gradient norm of 1.0 [39]. At the end of each epoch, a snapshot of the models' parameters was saved. The final parameters chosen for each model were those among the snapshots that yielded the highest validation accuracies. Both models were able to achieve high training accuracies, with *DeepConvLSTM* achieving nearly perfect training accuracies, but such high training accuracies occurred at the expense of validation accuracy (see Fig. 10). Validation accuracy here refers to the accuracy of the model measured on a dataset not used for training. A dataset can be considered a statistical distribution. When evaluating

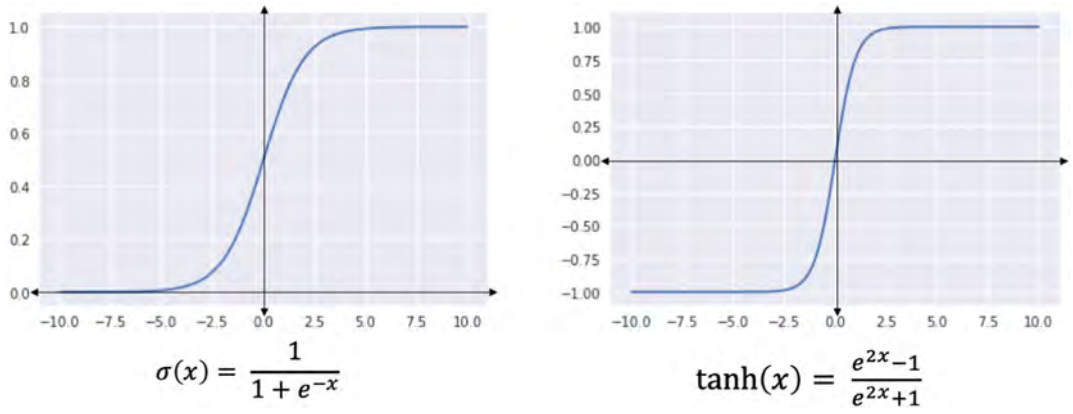

Fig. 8. The equations and graphs of the sigmoid and tanh functions.

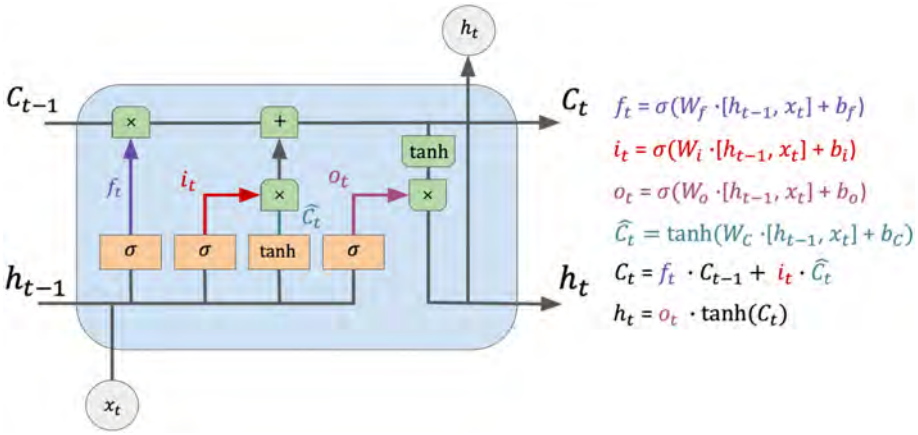

Fig. 9. A diagram of an LSTM cell and its governing equations. Inspired by Olah [37].

machine learning models, it is a standard practice to split a dataset into two or more disjoint subsets while attempting to retain the statistical characteristics of the overall distribution. A model trained on a subset of the data called the training set can be evaluated on another subset of the data that it did not encounter during training called the validation set. Validation accuracy in the context of this work is defined as the ratio of correct activity label predictions to the total number of activity labels produced by a model observing the validation set after being trained on the training set.

Although one might worry than an LSTM that tends to overfit the training data is simply memorizing the training data, the highest validation accuracy *DeepConvLSTM* achieved was better than the highest validation accuracy that *BaselineCNN* managed, suggesting that *DeepConvLSTM* retained significant predictive value beyond mere memorization.

5.2. Classification overview

Although *BaselineCNN* and *DeepConvLSTM* both managed reasonably good validation accuracies in classifying the compactor's activities, *DeepConvLSTM* displayed higher performance. When dealing with easier subproblems where similar activities were combined into a single category, the performance of both models improved. As *DeepConvLSTM* was shown to be superior in identifying the roller compactor's activities across all trials, it was the only model applied to the excavator. This is because the goal of the comparison is to find a deep learning model that has a high accuracy in predicting construction equipment activities which can be used universally for all the equipment types. In the excavator experiment, *DeepConvLSTM* achieved high validation accuracies, making mainly reasonable errors despite the excavator's activities being more complex than the roller compactor's activities.

5.3. Compactor: Six-activity identification problem

Tasked with distinguishing among six activities of the roller compactor, *BaselineCNN* managed a respectable validation accuracy of 74.2% and *DeepConvLSTM* went further, achieving a validation accuracy of 77.1%. Additional performance metrics including precision, recall, and F1 score for both models are presented in Table 1. To illustrate how the models might behave when modeling real-time equipment activities, the predictions of both *BaselineCNN* (a) and *DeepConvLSTM* (b) are plotted against the ground truth activity labels in Fig. 11. Deviations from the ground truth signal plotted as a bold black line present as spikes of blue or green on the plots. Overall, the *DeepConvLSTM* predictions displayed in blue match the ground truth signal much more closely than the *BaselineCNN* predictions shown in green.

5.4. Compactor: Direction-only subproblem

With the possible activity labels reduced to just *Forward* and *Backward* by combining all of the vibration settings with the same direction into a single class, *BaselineCNN* achieved a high validation accuracy of 93.6% and a high average F1 score of 0.94. *DeepConvLSTM* achieved an even higher validation accuracy of 96.2% and a similarly higher average F1 score of 0.96. Distinguishing the direction in which the roller compactor was moving was very well handled by both models.

5.5. Compactor: Vibration-setting only subproblem

With the possible activity labels reduced to just the vibration settings *High, Low,* and *Off* by ignoring the direction in which the roller compactor was moving, *BaselineCNN* managed a reasonable validation

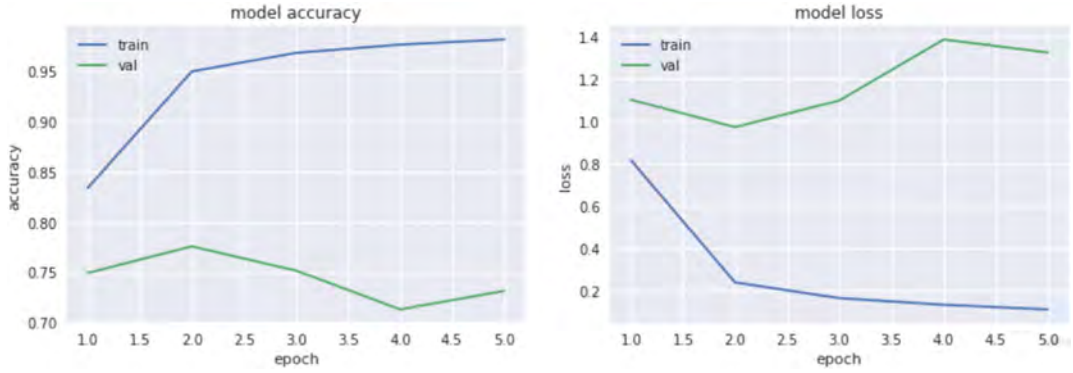

Fig. 10. Accuracy and loss curves for *DeepConvLSTM* in the six-class compactor experiment.

Table 1
Compactor activity metrics for *BaselineCNN* and *DeepConvLSTM*.

Activity label	Precision		Recall		F1-Score	
	BaselineCNN	*DeepConvLSTM*	*BaselineCNN*	*DeepConvLSTM*	*BaselineCNN*	*DeepConvLSTM*
Forward (Vibration: High)	0.73	0.81	0.77	0.73	0.75	0.77
Backward (Vibration: High)	0.81	0.75	0.34	0.32	0.47	0.45
Forward (Vibration: Low)	0.65	0.72	0.67	0.8	0.66	0.76
Backward (Vibration: Low)	0.76	0.75	0.91	0.93	0.83	0.83
Forward (Vibration: Off)	0.87	0.80	0.72	0.9	0.79	0.85
Backward (Vibration: Off)	0.69	0.86	0.99	0.86	0.82	0.86
Average	0.75	0.78	0.73	0.76	0.72	0.75

accuracy of 74.4% and an average F1 score of 0.75. *DeepConvLSTM* achieved a slightly higher validation accuracy of 75.2% and a slightly higher average F1 score of 0.75. Distinguishing the vibration mode in which the compactor was operating appears to be a much harder sub-problem than distinguishing the directions of its movements.

5.6. Excavator: Seven-activity identification problem

In this problem, possible activities were *Idling, Traveling, Scooping, Dropping, Rotating (left), Rotating (right),* and *Various*. *DeepConvLSTM* achieved a validation accuracy of 77.6% and an average F1 score of 0.78. Although the dataset was imbalanced in favor of the *Various* activity class (over 40% of the data), counteracting the imbalance with a weighted loss function decreased the F1 score. The unweighted results

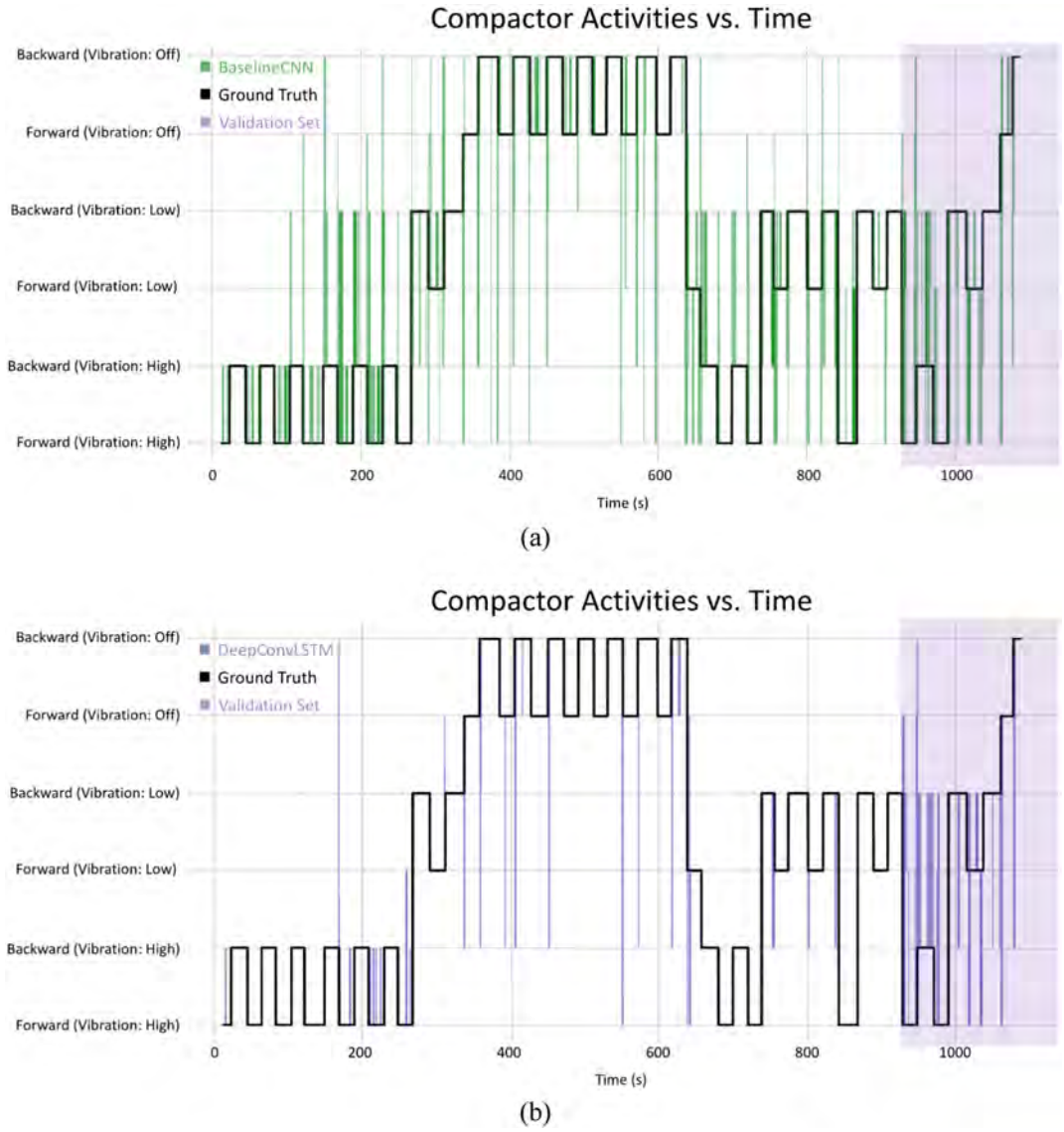

Fig. 11. (a) Compactor activity predictions of *BaselineCNN* vs. the ground truth data and (b) Compactor activity predictions of *DeepConvLSTM* compared to the ground truth data.

Table 2
Excavator activity metrics for *DeepConvLSTM*.

Activity label	Precision			Recall			F1-Score		
	Full data	No Various	Adjusted Idle	Full data	No Various	Adjusted Idle	Full data	No Various	Adjusted Idle
Idling	0.90	1.00	1.00	0.97	0.96	0.81	0.93	0.98	0.89
Traveling	0.42	0.99	0.99	0.22	0.57	0.57	0.29	0.72	0.72
Scooping	0.32	0.70	0.73	0.75	0.96	0.96	0.45	0.81	0.83
Dropping	0.66	0.83	0.83	0.65	0.65	0.65	0.66	0.73	0.73
Rotating (left)	0.68	0.69	0.69	0.74	0.93	0.93	0.71	0.79	0.79
Rotating (right)	0.82	0.94	0.94	0.80	0.80	0.80	0.81	0.86	0.86
Various	0.84	N/A	N/A	0.65	N/A	N/A	0.73	N/A	N/A
Average	0.81	0.92	0.85	0.78	0.91	0.83	0.78	0.91	0.83

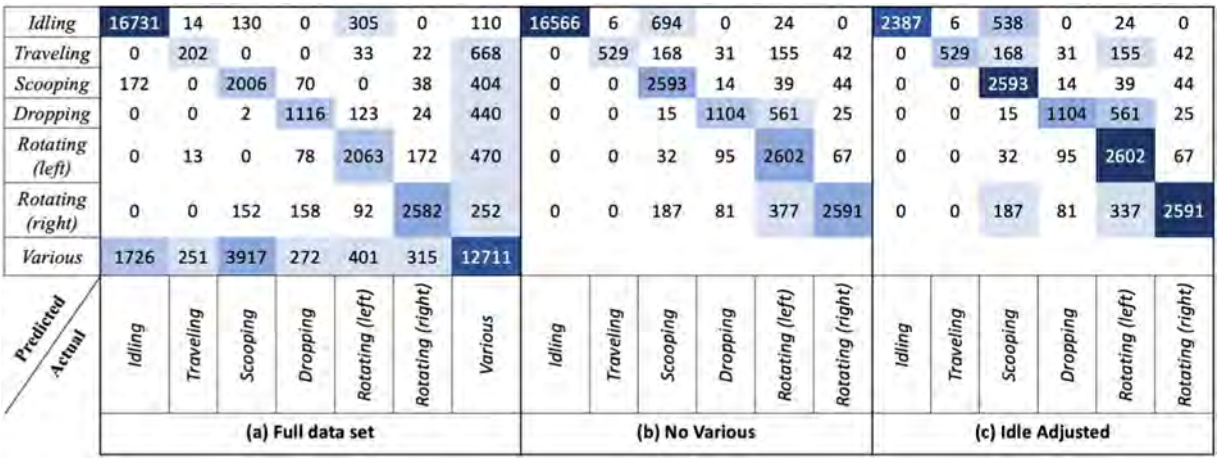

Fig. 12. The confusion matrices for *DeepConvLSTM's* performance in the excavator case. Predicted labels on the vertical axis; actual labels on the horizontal axis.

were judged to be most representative and are summarized in Table 2. This is not surprising since *Various* consists of multiple actions from the other categories rather than being a distinct activity itself. The model struggled a little in identifying the *Traveling* activity, but it only comprised 2% of the data. As the confusion matrix in Fig. 12(a) shows, most of the model's errors were related to the *Various* activity. To illustrate the model's predictive power beyond confusion related to the *Various* category, two additional sets of performance metrics are reported (see Table 2 and Fig. 12).

The *No Various* and *Adjusted Idle* results derive from an instance of *DeepConvLSTM* trained and evaluated separately on a subset of the full data set. To compose this subset, every frame with the label *Various* was omitted from both training and validation. This setup is somewhat artificial since it renders the model incapable of reasonably processing the full data set as it is. In other words, it would not know what to do with all of the *Various* labels since that category is no longer in its vocabulary. However, it provides a reasonable estimation of how the model might perform in scenarios where there is no ambiguous label like *Various* – after all, this label is merely an artifact of the difficulties of manually labeling the ground truth data when many complex activities are involved. *DeepConvLSTM* managed a very high validation accuracy of 90.7%, and an average F1 score of 0.91. As the confusion matrix in Fig. 12(b) suggests, the model benefitted somewhat from the fact that the removal of the *Various* activities left a disproportionately high number of *Idle* frames. Intermediate results throughout experimentation suggest that the *Idle* activity is fairly easy to classify with extremely high accuracy. To give an estimate of the model's performance under conditions that are less favorable but still unambiguous, the same model (trained without the *Various* frames) was evaluated on a modified version of its validation set with the first 14,335 instances of *Idle* removed as well (*Adjusted Idle*). Under these conditions, the class distribution in the validation data set was well-balanced. *DeepConvLSTM* still managed a respectable validation accuracy of 82.5% and an average F1 score of 0.83.

6. Discussion

As illustrated above in Fig. 10, the models were able to fit the training data almost perfectly, particularly *DeepConvLSTM*. Although this work did not manage to translate that performance perfectly to the validation set, that level of training set performance suggests that the models are complex enough to predict the activities of construction equipment from time series of accelerometer data recorded during the equipment's activities. Despite the large amount of sensor readings available due to the relatively high sampling rate of 100 Hz, the data sets used in these experiments only represented 20 to 30 min of real-world activities. It is expected that the models, in particular *DeepConvLSTM*, would better overcome the generalization gap given more training data. Furthermore, state-of-the-art results in human gesture recognition benefitted greatly from combining different sensor modalities [32]. In the cases presented here, only limited sensor data was available beyond the accelerometer information, and preliminary attempts to leverage those additional sensor modalities did not improve the results. It is likely that additional kinds of sensor information could improve the accuracies of the models' activity predictions; however, it is also encouraging that accelerometer data are a potent source of information on construction equipment activities by themselves.

At 74.2%, the validation accuracy that *BaselineCNN* achieved in the full roller compactor activity classification problem appears similar to the validation accuracy of 77.1% that *DeepConvLSTM* achieved. However, as the plots of the models' predictions against the ground truth data shown in Fig. 11 highlight, *DeepConvLSTM* managed much smoother predictions with a character qualitatively similar to that of the ground truth data (i.e. the predictions have fewer jagged jumps

deviating from the ground truth), revealing it to be a significantly superior model. This visual quality of smoothness is closely related to the quantitative measurement *accuracy*, which is why the model's predictions are particularly smooth on the training set, where it achieved almost 100% accuracy. Should it be necessary for further study, this notion of "smoothness" could be more directly quantified by a measure such as the number of activity label changes per some standardized number of time steps.

In addition to showing significantly smoother predictions in the case of the compactor, *DeepConvLSTM* was also able to accurately predict the activities of the excavator. As the confusion matrix in Fig. 12a above shows, many of *DeepConvLSTM's* mistakes appear to come from plausible similarities between the *Various* activity category and isolated activities overlapping with it. As recorded in Table 2, *DeepConvLSTM* achieved very high performance when the ambiguity of the *Various* class was removed from its considerations. In addition to showing significantly smoother predictions in the case of the compactor, *DeepConvLSTM* was also able to accurately predict the activities of the excavator. As the confusion matrix in Fig. 12a above shows, many of *DeepConvLSTM's* mistakes appear to come from plausible similarities between the *Various* activity category and isolated activities overlapping with it. As recorded in Table 2, *DeepConvLSTM* achieved very high performance when the ambiguity of the *Various* class was removed from its considerations.

It seems a reasonable conjecture that the movements of construction machinery are less likely to be peculiar to individual subjects than the movements of human workers since equipment is made in standardized shapes and sizes and articulates in more prescribed ways. Thus, models trained to recognize the activities of a single piece of equipment or a small set of machines should retain accuracy when tasked with predicting the activities of different machines of the same kind. Any equipment activity recognition system released in production will require this property to apply the training it received in the factory to the customer's equipment. Future work will study how well models generalize to different machines of a given type after being trained on particular machines of that type, as well as study the links between machine activities and their emissions.

In addition, data augmentation techniques applicable to time series such as the jittering, scaling, rotation, and time-warping very recently explored with great success by Rashid and Louis [15] seem likely to improve the accuracies of deep learning models trained on accelerometer data, particularly when the amount of available training data is small. It is interesting that the results presented here show significantly higher validation accuracies when no data augmentation is used (around 80% compared to around 60% in the most comparable work of Rashid and Louis), especially because the work here is based on much less available data (145,068 time steps on average across the pieces of equipment studied here compared to approximately 576,000 time steps in the work of Rashid and Louis). A significant difference that could explain the significantly better accuracies found here under the aforementioned conditions is the use of convolutional feature extraction in this study. Future work should more carefully study the impacts of convolutional feature extraction and of data augmentation in time-series-based deep learning models. It seems likely a combination of these two techniques could yield better results.

7. Limitations

While the presented study is a promising early step toward building automatic, reliable, and extendible construction equipment activity recognition frameworks, even better results can be obtained if certain limitations are addressed. First of all, the authors cannot eliminate the possibility that the models used could be improved in some way – either by fixing some undetected flaw or by using some superior architecture. However, given that nearly perfect training accuracy was achieved and that additional regularization could not reduce the gap between training and validation performance, the authors are left to conclude that the models suffered most significantly from not having enough data. Although the data augmentation techniques explored by Rashid and Louis [15] could help, the best solution would be gathering more data. The volume of data studied here was the maximum amount the researchers could get access to in a timely manner given a preference for real data in a live environment and restrictions on collecting data on the industry partner's jobsite. Even though this dataset was large enough to show the promise of the models developed, real-world operations of construction equipment entail a great deal of variations within an activity that would not allow models to distinguish them from the variations that denote changes in activities, if the dataset is not considerably large. Researchers working in construction equipment activity recognition would benefit from assembling a large, open body of quality data on multiple types and multiple instances of equipment performing various activities that would allow stronger predictive claims to be made at the level required by future production activity recognition systems.

8. Conclusion

Automated analysis of heavy construction equipment activities provides insight into the performance of the involved resources. The work here proposed a framework for automated analysis of heavy construction equipment activities involving outfitting the equipment with low-cost accelerometer sensors, labeling the activities from real work performed while recording acceleration patterns produced by the equipment, and training deep learning models to make predictions about the equipment's activities when labels are not available. This framework was evaluated using two different deep learning architectures – *BaselineCNN* and *DeepConvLSTM* – on two different kinds of heavy machinery performing very different tasks – a roller compactor landscaping at a hotel and an excavator digging a sewage trench. The MyoMotion 684 sensors used in these experiments provided more reliable and more accurate readings free of calibration issues that previous work faced when using sensors embedded in smartphones. Unlike previous activity recognition frameworks relying on shallow machine learning models, the deep learning architectures used here were able to extract features by themselves, eliminating a great deal of manual work that would otherwise be spent on feature selection, achieving greater accuracies without the biases introduced by manual feature design, and allowing the models to adapt to the different kinds of equipment studied without manual modification across cases.

Whereas previous research into construction equipment activity recognition has largely only considered a few broad activity categories like *Idling, Working,* and *Off*, the models presented here were able to achieve very high accuracies even while distinguishing between much more complex activity categories. Given that these models are dependent only on accelerometer data, it should be relatively easy to outfit equipment with the necessary sensors. The best model studied, *DeepConvLSTM,* reached particularly high validation accuracies despite having access to limited training data. Since accelerometers are not intrusive, it is feasible that an activity monitoring system reliant on them could be deployed without disrupting normal work; from there, the system could continuously gather data and improve its performance.

Previous studies in this area of research have established a compelling precedent on the importance of construction activity recognition and its implications on improving productivity, safety, and sustainability measures of construction projects. This study contributes to the construction engineering and management body of knowledge by advancing this research topic in various ways. As a first step toward the important goal of creating automatic, scalable, and adaptable equipment activity recognition, this study advances the research in this area by showing how a hybrid deep learning model consisting of both convolutional and LSTM layers can reliably predict equipment activities. As

indicated in the Limitations section, obtaining more training data could push accuracy and reliability higher, even with the same architectures. Armed with accurate activity recognition models, future studies could make claims linking patterns in activity transitions to greater emissions or lower productivity, which in turn increases emissions released to complete a given construction task due to increased equipment running times. Models tracking these quantities on an entire fleet after having been trained on similar equipment by researchers or manufacturers could suggest specific modifications to operators' working styles and work patterns to lower emissions, improve productivity, or both at scale.

Moreover, the presented research is part of an ongoing study that aims to predict emission levels resulting from the activities of heavy construction equipment. Building upon the findings of this work, the authors are working on enhanced deep learning architectures that enable mapping equipment activities to the emission levels of various gases with environmental impacts.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The presented work is supported by the California Senate Bill 1 (SB1): California State University Transportation Consortium (CSUTC), grant #1852. The authors gratefully acknowledge CSUTC for their financial support. The authors would also like to thank Guerra Construction Group for allowing access to their job sites for data collection. Any opinions, findings, conclusions, and recommendations expressed in this paper are those of the authors and do not necessarily represent those of the CSUTC or Guerra Construction Group.

References

[1] S. Mansouri, F. Castronovo, R. Akhavian, Analysis of the synergistic effect of data analytics and technology trends in the AEC/FM industry, J. Constr. Eng. Manag. 146 (3) (2020) 04019113, , https://doi.org/10.1061/(ASCE)CO.1943-7862.0001759.

[2] Plangrid, Construction disconnected. New research from PlanGrid and FMI identifies factors costing the construction industry more than $177 billion annually, https://www.plangrid.com/press/fmi/, (2019).

[3] G. Ballard, G. Howell, What is lean construction, Seventh Conference of the International Group for Lean Construction, California-USA, IGLC, Paper, 7 1999.

[4] T. Cheng, J. Teizer, G.C. Migliaccio, U.C. Gatti, Automated task-level activity analysis through fusion of real time location sensors and worker's thoracic posture data, Autom. Constr. 29 (2013) 24–39, https://doi.org/10.1016/j.autcon.2012.08.003.

[5] R. Akhavian, A.H. Behzadan, Productivity analysis of construction worker activities using smartphone sensors, *Proceedings of the 16th International Conference Computing in Civil Building Engineering*, 2016, pp. 1067–1074 (Osaka, Japan, ISBN 978-4-9907371-2-2).

[6] C.F. Cheng, A. Rashidi, M.A. Davenport, D.V. Anderson, Activity analysis of construction equipment using audio signals and support vector machines, Autom. Constr. 81 (2017) 240–253, https://doi.org/10.1016/j.autcon.2017.06.005.

[7] R.M. Wideman, A pragmatic approach to using resource loading, production, and learning curves on construction projects, Can. J. Civ. Eng. 21 (6) (1994) 939–953, https://doi.org/10.1139/194-100.

[8] Bureau of Labor Statistics, Measuring Productivity Growth in Construction, U.S. Bureau of Labor Statistics, 2018, https://www.bls.gov/opub/mlr/2018/article/measuring-productivity-growth-in-construction.htm (accessed 11.7.18).

[9] J. Gong, C.H. Caldas, An object recognition, tracking, and contextual reasoning-based video interpretation method for rapid productivity analysis of construction operations, Autom. Constr. 20 (8) (2011) 1211–1226, https://doi.org/10.1016/j.autcon.2011.05.005.

[10] R. Gravina, P. Alinia, H. Ghasemzadeh, G. Fortino, Multi-sensor fusion in body sensor networks: state-of-the-art and research challenges, Information Fusion 35 (2017) 68–80, https://doi.org/10.1016/j.inffus.2016.09.005.

[11] J. Wu, A. Akbari, R. Grimsley, R. Jafari, A decision level fusion and signal analysis technique for activity segmentation and recognition on smart phones, Proceedings of the 2018 ACM International Joint Conference and 2018 International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers, ACM, 2018, pp. 1571–1578, , https://doi.org/10.1145/3267305.3267525.

[12] R. Akhavian, A.H. Behzadan, Construction equipment activity recognition for simulation input modeling using mobile sensors and machine learning classifiers, Adv. Eng. Inform. 29 (4) (2015) 867–877, https://doi.org/10.1016/j.aei.2015.03.001.

[13] J. Cai, Y. Zhang, H. Cai, Two-step long short-term memory method for identifying construction activities through positional and attentional cues, Autom. Constr. 106 (2019) 102886, , https://doi.org/10.1016/j.autcon.2019.102886.

[14] J. Kim, S. Chi, Action recognition of earthmoving excavators based on sequential pattern analysis of visual features and operation cycles, Autom. Constr. 104 (2019) 255–264, https://doi.org/10.1016/j.autcon.2019.03.025.

[15] K.M. Rashid, J. Louis, Times-series data augmentation and deep learning for construction equipment activity recognition, Adv. Eng. Inform. 42 (2019) 100944, , https://doi.org/10.1016/j.aei.2019.100944.

[16] R. Akhavian, A.H. Behzadan, Remote monitoring of dynamic construction processes using automated equipment tracking, Construction Research Congress 2012: Construction Challenges in a Flat World, 2012, pp. 1360–1369, , https://doi.org/10.1061/9780784412329.137.

[17] L. Joshua, K. Varghese, Accelerometer-based activity recognition in construction, J. Comput. Civ. Eng. 25 (5) (2011) 370–379, https://doi.org/10.1061/(ASCE)CP.1943-5487-0000097.

[18] M. Golparvar-Fard, A. Heydarian, J.C. Niebles, Vision-based action recognition of earthmoving equipment using spatio-temporal features and support vector machine classifiers, Adv. Eng. Inform. 27 (4) (2013) 652–663, https://doi.org/10.1016/j.aei.2013.09.001.

[19] J.M. Monnot, R.C. Williams, Construction equipment telematics, J. Constr. Eng. Manag. 137 (10) (2011) 793–796, https://doi.org/10.1061/(asce)co.1943-7862.0000281.

[20] Walt Moore, Practical telematics, Construction Equipment, 2012 https://www.constructionequipment.com/practical-telematics , Accessed date: 10 December 2019.

[21] T.L. Jones, C. Schlegel, Can real time location system technology (RTLS) provide useful estimates of time use by nursing personnel? Research in Nursing & Health 37 (1) (2014) 75–84, https://doi.org/10.1002/nur.21578.

[22] G. Judah, J. de Witt Huberts, A. Drassal, R. Aunger, The development and validation of a real time location system to reliably monitor everyday activities in natural contexts, PLoS One 12 (2) (2017) e0171610, , https://doi.org/10.1371/journal.pone.0171610.

[23] R. Akhavian, A.H. Behzadan, Evaluation of queuing systems for knowledge-based simulation of construction processes, Autom. Constr. 47 (2014) 37–49, https://doi.org/10.1016/j.autcon.2014.07.007.

[24] H. Jiang, P. Lin, Q. Fan, M. Qiang, Real-time safety risk assessment based on a real-time location system for hydropower construction sites, Sci. World J. 2014 (2014), https://doi.org/10.1155/2014/235970.

[25] C.R. Ahn, S. Lee, F. Peña-Mora, Application of low-cost accelerometers for measuring the operational efficiency of a construction equipment fleet, J. Comput. Civ. Eng. 29 (2) (2013) 4014042, https://doi.org/10.1061/(ASCE)CP.1943-5487.0000337.

[26] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (7553) (2015) 436–444, https://doi.org/10.1038/nature14539.

[27] J. Yang, M.N. Nguyen, P.P. San, X.L. Li, S. Krishnaswamy, Deep convolutional neural networks on multichannel time series for human activity recognition, Twenty-Fourth International Joint Conference on Artificial Intelligence, 2015, pp. 3995–4001 http://dl.acm.org/citation.cfm?id=2832747.2832806.

[28] J. Wang, Y. Chen, S. Hao, X. Peng, L. Hu, Deep learning for sensor-based activity recognition: a survey, Pattern Recogn. Lett. 119 (2019) 3–11, https://doi.org/10.1016/j.patrec.2018.02.010.

[29] S. Ren, K. He, R. Girshick, J. Sun, Faster r-cnn: Towards real-time object detection with region proposal networks, Advances in Neural Information Processing Systems, 2015, pp. 91–99, , https://doi.org/10.1109/tpami.2016.2577031.

[30] C. Hernandez, T. Slaton, V. Balali, R. Akhavian, A deep learning framework for construction equipment activity analysis, *Computing in Civil Engineering 2019: Data, Sensing, and Analytics*, Atlanta, GA, American Society of Civil Engineers, Reston, VA, 2019, pp. 479–486, , https://doi.org/10.1061/9780784482438.061.

[31] Noraxon USA, 2018. myoMotion research PRO IMU, https://www.noraxon.com/our-products/research-pro-imu/. (accessed 11.15.18).

[32] F. Ordóñez, D. Roggen, Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition, Sensors 16 (1) (2016) 115, https://doi.org/10.3390/s16010115.

[33] X. Shi, Z. Chen, H. Wang, D.Y. Yeung, W.K. Wong, W.C. Woo, Convolutional LSTM network: A machine learning approach for precipitation nowcasting, *Advances in Neural Information Processing Systems*, Montréal Canada, Neural Information Processing Systems Foundation, Inc, San Diego, CA, 2015, pp. 802–810 https://dl.acm.org/doi/10.5555/2969239.2969329 , Accessed date: 5 February 2020.

[34] R. Akhavian, A.H. Behzadan, Simulation-based evaluation of fuel consumption in heavy construction projects by monitoring equipment idle times, 2013 Winter Simulations Conference (WSC), IEEE, 2013, pp. 3098–3108, , https://doi.org/10.1109/wsc.2013.6721677.

[35] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting, The Journal of Machine Learning Research 15 (1) (2014) 1929–1958 http://jmlr.org/papers/v15/srivastava14a.html (accessed 2.5.2020).

[36] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780, https://doi.org/10.1016/neco.1997.9.8.1735.

[37] C. Olah, Understanding lstm networks, https://colah.github.io/posts/2015-08-Understanding-LSTMs/, (2015).

[38] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, International Conference on Learning Representations, 2014 https://*arXivpreprintarXiv:1412.6980* , Accessed date: 5 February 2020.

[39] R. Pascanu, T. Mikolov, Y. Bengio, On the difficulty of training recurrent neural networks, International Conference on Machine Learning, Atlanta, GA, 2013, pp. 1310–1318 https://arxiv.org/abs/1211.5063 , Accessed date: 5 February 2020.