Advanced Engineering Informatics 42 (2019) 100944

Contents lists available at ScienceDirect

Advanced Engineering Informatics

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

Full length article

Times-series data augmentation and deep learning for construction equipment activity recognition

Khandakar M. Rashid, Joseph Louis*

School of Civil and Construction Engineering, Oregon State University, Corvallis, OR 97331, USA

ARTICLE INFO

Keywords:
Construction equipment activity recognition
Inertial measurement unit
Deep learning
Time-series data augmentation
LSTM network
Big data analytics

ABSTRACT

Automated, real-time, and reliable equipment activity recognition on construction sites can help to minimize idle time, improve operational efficiency, and reduce emissions. Previous efforts in activity recognition of construction equipment have explored different classification algorithms anm accelerometers and gyroscopes. These studies utilized pattern recognition approaches such as statistical models (e.g., hidden-Markov models); shallow neural networks (e.g., Artificial Neural Networks); and distance algorithms (e.g., K-nearest neighbor) to classify the time-series data collected from sensors mounted on the equipment. Such methods necessitate the segmentation of continuous operational data with fixed or dynamic windows to extract statistical features. This heuristic and manual feature extraction process is limited by human knowledge and can only extract human-specified shallow features. However, recent developments in deep neural networks, specifically recurrent neural network (RNN), presents new opportunities to classify sequential time-series data with recurrent lateral connections. RNN can automatically learn high-level representative features through the network instead of being manually designed, making it more suitable for complex activity recognition. However, the application of RNN requires a large training dataset which poses a practical challenge to obtain from real construction sites. Thus, this study presents a data-augmentation framework for generating synthetic time-series training data for an RNN-based deep learning network to accurately and reliably recognize equipment activities. The proposed methodology is validated by generating synthetic data from sample datasets, that were collected from two earthmoving operations in the real world. The synthetic data along with the collected data were used to train a long short-term memory (LSTM)-based RNN. The trained model was evaluated by comparing its performance with traditionally used classification algorithms for construction equipment activity recognition. The deep learning framework presented in this study outperformed the traditionally used machine learning classification algorithms for activity recognition regarding model accuracy and generalization.

1. Introduction

The construction industry is recognized to have a lower productivity when compared to other industries that produce engineered products like manufacturing [1]. One major contributing factor is the temporary and transient nature of most operations, which make it difficult to implement systems that collect and analyze data to provide insights into that operation. Thus, one of the steps towards productivity improvement is to improve the techniques of assessing and monitoring the performance of key resources. Owning and maintaining heavy construction equipment contributes to a large portion of the total project costs, especially in the case of heavy civil projects. Therefore, identifying and tracking their activities plays an important role in measuring their performance, which itself is the primary pre-requisite to enable performance improvement. Automated, real-time, and reliable activity recognition of heavy construction equipment is thus a necessary step that enables several other practical applications such as automated cycle-time analysis [2–4], productivity monitoring [5–7], safety applications [8–11], environmental assessment [12,13], near real-time simulation inputs [4,14,15], and applications in AR/VR visualization [16–18]. To develop an effective activity recognition framework for construction equipment, several previous studies have explored the feasibility of using location data and/or time-series vibration data from inertial measurement units (IMU) [2,13,14].

Even though these studies have shown promising results, traditional machine learning approaches require manual segmentation of time-series data after collecting the training data, and further extraction of statistically significant feature vectors. The quality of features in these

* Corresponding author.
E-mail addresses: rashidk@oregonstate.edu (K.M. Rashid), joseph.louis@oregonstate.edu (J. Louis).

https://doi.org/10.1016/j.aei.2019.100944
Received 1 February 2019; Received in revised form 28 April 2019; Accepted 7 June 2019

methods depends on the human domain knowledge as this approach can only extract human-specified shallow features. Moreover, traditional machine learning approaches treat each time step of the time-series sensor data statistically independently, not accounting the temporal relationship between consecutive time steps. In contrast, deep learning methods, such as recurrent neural networks (RNN) are able to automatically extract high-level representative features that consider temporal dynamics among conseutive time steps of the sensor data. This deep learning approach can successfully obviate the need for manual data segmentation and feature extraction process of traditional shallow by offering higher representational power. However, training a deep network typically requires large training dataset which poses a practical challenge to obtain from construction equipment in the real-world due to its (aforementioned) temporary and transient nature. Data augmentation techniques can be implemented to generate synthetic training data to address this issue of data paucity. Data augmentation enhances the available limited dataset by transforming existing samples to create new ones [19]. This approach of augmenting training data is a well-known methodology in computer-vision domain but has not been fully explored in addressing time-series classification.

In this paper, the problem of identifying construction activities is tackled using long-short time memory (LSTM) network trained with data captured by IMUs from articulated elements of the equipment. To train the LSTM network with sufficient training data, time-series data augmentation techniques are developed and implemented to generate a synthetic training dataset. The performance of the LSTM network is compared with the traditional shallow network (i.e. artificial neural network (ANN)). The effect of time-series data augmentation techniques on the model performance is evaluated using field data from an excavator and a front-end loader. The framework developed in this study can be extended to any articulated equipment used in construction such as loader, excavator, and articulated trucks etc.

This paper is organized by first presenting a review of the state-of-the-art in IMU-based equipment activity recognition, deep learning-based activity recognition, and time-series data augmentation methods to provide the context for the presented work. The identified research gaps, overall research goal, and objectives are then explicitly stated for the paper. Then, the methodology section discusses the technical details of the deep learning network and data augmentation techniques. Two different case studies, using an excavator and a front-end loader, are conducted to validate the developed framework. Finally, the results and main contributions of this study is summarized along with limitations and future research directions.

2. Background and related work

The framework presented in this paper consists of a deep learning-based activity recognition framework for construction equipment using multiple IMUs; and time-series data augmentation techniques to generate synthetic training data. This section thus provides a comprehensive literature review of construction equipment activity recognition, deep learning for sensor-based activity recognition, and data augmentation techniques.

2.1. Construction equipment activity recognition

Activity identification methods for construction equipment can be broadly classified into vision-based and sensor-based methods. In the vision-based domain, Zou and Kim [7] used image processing to quantify the idle-times of hydraulic excavators. This study was limited to identifying only two states of the excavator: idle and busy. Azar and McCabe [20] proposed an activity recognition framework using rational events to recognize dirt-loading activities of an excavator. Bao et al. [21] investigated the use of long-sequence videos to automatically detect, track, and identify activities of an excavator and a dump truck. In a similar effort, excavator and dump trucks were also used to measure the performance of earthmoving operations utilizing image frame sequences [22]. The concept of bag-of-video-feature-words model was extended using unsupervised classifiers into the construction domain to learn and classify labor and equipment activities [23]. Vision-based techniques have shown promising results in tracking construction resources and identifying their operations. However, these techniques provide very limited information based on the field of view of the cameras used. It is a challenging to maintain a direct line of sight to targeted resources due to high level of noises (e.g., entity overlap, moving backgrounds, varying light conditions etc.) on dynamic construction sites. These challenges can be overcome by adopting motion sensors which are not so constrained.

Contrary to vision-based methods, sensor-based approaches leverage the location and/or vibration of the equipment to identify its activity. El-Omari and Moselhi et al. [24] and Ergen et al. [25] proposed a framework combining radio frequency identification (RFID) and global positioning system (GPS) technology for automated location and tracking of construction equipment. Vahdatikhaki and Hammad [4] enhanced the performance of equipment state-identification by adopting a multi-step data processing framework combining location and motion data. Song and Eldin [26] developed an adaptive real-time tracking of equipment based on location to improve look-ahead scheduling accuracy.

Although location-based operation tracking can identify the state of construction equipment at a coarse level (e.g.: *idle* and *busy* states), it is incapable of classifying the activities at a finer level of detail, especially when the equipment is performing activities from a stationary position. Such limitations of location-based operation tracking have inspired the exploration of both independent [13] and smartphone-embedded [9,10] inertial measurement units (IMUs) for activity recognition. Ahn et al. [13] used a low-cost accelerometer mounted inside the cabin of an excavator to collect data of an earthmoving worksite to classify three different states (i.e. engine-off, idle, and busy) of an excavator. Mathur et al. [2] utilized smartphone-embedded accelerometer by mounting it inside an excavator cabin to measure various activity modes (e.g. wheel base motion, cabin rotation, and arm movement) and duty cycles. Akhavian and Behzadan [14] adopted a similar approach by attaching a smartphone to the cabin of a front-end loader to collect accelerometer and gyroscope data during an earthmoving operation, upon which several classification algorithms (i.e., ANN, DT, KNN, LR, SVM) were tested. Their study also investigated the impact of different technical parameters such as level of detail, segmentation window size, and selection of features on the performance of different classification algorithms. The same approach were further extended for construction workers [27].

Even though significant advancements have been achieved in activity recognition for construction equipment using RTLS/IMU sensors, most of those studies implement pattern recognition approaches. These approaches require segmentation of continuous time-series data with fixed and/or dynamic windows to extract statistical features. This heuristic and manual feature extraction process is limited by human domain knowledge and only can learn using human-specified shallow features. Moreover, pattern recognition methods treat each time step of the sensor data statistically independently, ignoring the temporal dynamics between them. These limitations of existing methods motivate this research that uses deep learning to automatically extract features containing temporal dependency, from sensor data. A review of deep learning implementations in the activity recognition domain is provided in the next section.

2.2. Deep learning for sensor-based activity recognition

Extensive research studies have been conducted that implement deep learning algorithms to develop activity recognition frameworks, especially in the area of human activity recognition (HAR). Some of the most common types of deep learning models used in activity

recognition tasks are deep neural network (DNN), convolutional neural network (CNN), recurrent neural network (RNN), deep belief network (DBN), and stacked autoencoder (SAE). Vepakomma et al. [28] used a DNN model to identify indoor activities of elderly people. In doing so, hand-engineered features were extracted from the wrist-worn sensors and then those features were fed into a DNN. In another effort, Walse and Dhakaskar [29] performed principal component analysis (PCA) before feeding the features to DNN model. In these efforts, authors only used DNN as a classification model after hand-crafted feature extraction, which may not generalize the model optimally, and which may cause a shallow network. To improve the performance of DNN, researchers used higher number of hidden layers to automatically extract features and generalize the deep network [30]. Improved performance of automated feature extraction using more hidden layers indicates that when HAR data are multi-dimensional and activities are more complex, more hidden layers can help train the deep network by strengthening their representation capability [31]. Convolutional neural network (CNN) is another deep learning model which is capable of automatic extraction of features from signals and it has achieved promising results in the HAR domain. In an earlier work, each dimension of the sensor was treated as one channel (like RGB in an image), and then the convolution and pooling were performed separately [32]. In another study, a CNN framework was proposed to automate feature learning from the raw input to unify and share weights in multi-sensor data using 1D convolution [33]. This approach allowed higher level abstract representation of low-level time-series signals. In another similar work, passive RFID data were directly feed into deep CNN for activity recognition instead of selecting features and using a cascade structure that first detects object from RFID data followed by predicting the activity.

These de-facto standard approaches of activity recognition treat individual dimensions of the sensor data statistically independently. Thus, each dimension of the data is converted into feature vectors without due consideration of their broader temporal context. To address this, recurrent neural network (RNN) incorporates temporal dependencies of sensor data streams, which is more appropriate for activity recognition than considering the data stream independently. Long-short term memory (LSTM) cells are often incorporated with RNN, serving as the memory units through the gradient descent steps of the RNN. Inoue et al. [34] proposed a deep RNN-based activity recognition framework from raw accelerometer data, and investigated various architectures of the model to find the best parameter values. Ordóñez and Roggen [35] developed an LSTM-based RNN for multimodal wearable activity recognition which can perform sensor fusion naturally, does not require expert knowledge in designing features, and explicitly models the temporal dynamics of the feature vectors. This framework outperformed the previous results by up to 9%. Hammerla et al. [30] explored RNN approach for wearable activity recognition by introducing a novel regularization approach, and illustrated that the developed model outperformed the state-of-the-art non-recurrent approaches on a large benchmark dataset.

It can be seen that RNN, specifically LSTM networks, have the capability of modeling sequential time-series data by automatically extracting high-level representative features, and considering temporal relationship among each time step of the sensor data. This holds a lot of promise for its application in construction equipment activity recognition. However, training an LSTM network requires a large training dataset which is a practical challenge to obtain from the construction equipment in the real-world. This limitation is addressed by generating a large volume of synthetic training data from a smaller amount of collected data using data augmentation techniques that will be reviewed in the next section.

2.3. Data augmentation in shallow/deep learning

Data augmentation is a technique that enhances a limited amount of data by transforming the existing samples to create new data [19]. This technique has been implemented to generate synthetic training data in the computer vision [36–40], speech recognition [41–43], and time-series classification [19,44,45] domains. Charalambous and Bharath [38] introduced a simulation-based methodology which can be used for generating synthetic video data and sequence for machine/deep learning gait recognition algorithms. D'Innocente et al. [36] proposed an image data augmentation technique which zooms on the object of interest in an image and simulates the object detection outcome of a robot vision system to bridge the gap between computer and robot vision. Most advanced object recognition algorithms utilize various image augmentation techniques, such as flipping, rotating, scaling, cropping, translating, and adding Gaussian noise to generate synthetic data for training and testing machine/deep learning algorithms [39,40,46]. In the speech recognition domain, studies have investigated vocal tract length normalization [41,42], speech rate, and frequency-axis random distortion [41], label-preserving audio transformation [43] and evaluated those methods improve learning algorithms.

Despite the frequent implementation of data augmentation techniques in computer vision and speech recognition domain, data augmentation in time-series classification has not been deeply investigated yet [19]. Guennec et al. [47] proposed two time-series data augmentation techniques: window slicing and time-warping, to train a convolutional neural network (CNN). Forestier et al. [45] introduced dynamic time-warping (DTW) for time-series classification to reduce the variance of a classifier. Um et al. [19] proposed the most comprehensive set of time-series data augmentation techniques to monitor the Parkinson patient using wearable sensors. This research will add to the body of knowledge on time-series data augmentation by applying more transformations and by using it to enable activity recognition for construction equipment.

3. Research gaps and point of departure for this research

As can be observed from the literature review performed, this research targets the intersection of three specific research gaps from the domains of construction equipment activity identification, deep learning application, and time-series data augmentation. The specific research gaps addressed by this paper are stated below:

1. Existing methods that apply pattern recognition approaches towards identifying equipment activity from time-series sensor data require manual extraction of statistical features, which is a time-consuming process, and limited by the need for domain-specific knowledge. Moreover, these approaches do not consider the long-term temporal dependency between time steps of the time-series data. These limitations could be eliminated by automatically extracting highly representative features, containing the temporal dynamics of the sensor data using deep learning models.
2. Deep learning techniques have hitherto not been used in IMU-based construction equipment activity recognition due to the practical challenges posed by the requirement for large amount of training data. This research will address this limitation by synthesizing data using augmentation techniques.
3. Time-series data augmentation techniques is a relatively new area of research. This paper will add to the literature in this domain by utilizing more transformations and by providing a practical application of its techniques.

By considering the above specific gaps in literature, the overall goal for this paper is framed as enabling activity identification for construction equipment through the use of an LSTM network trained with synthetic data. This research goal is accomplished in this paper through the pursuit of three specific research objectives:

(1) Develop a deep learning activity recognition framework for

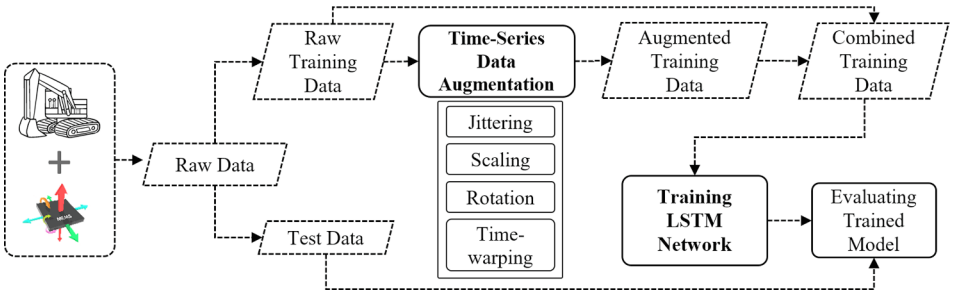

Fig. 1. Overall methodological architecture of this study.

construction equipment using motion data of articulated elements (e.g. bucket, boom, arm etc.) of the equipment.

(2) Develop time-series data augmentation techniques to generate synthetic training dataset to train the LSTM network with a large volume of training for better performance.

(3) Evaluate the performance of the LSTM network by comparing with traditional shallow networks (e.g. artificial neural network (ANN)) and determine the impact of data augmentation techniques on the performance of the training models.

4. Methodology

The methodological framework presented in this paper targets construction equipment with articulated parts such as earthmoving equipment (excavators, loaders, trucks, scrapers, etc.) and cranes (various types). The presence of moving parts on these equipment enable the use of motion sensors such as accelerometers and gyroscope to monitor the activities performed by these equipment. Fig. 1 illustrates the overall methodology used in this study to achieve the overarching research goal.

First, raw sensor data are collected from IMUs mounted on different articulated elements of construction equipment. Then, test data are separated from the raw data to check unbiased performance of the LSTM network with augmented training data. The remaining data, (*Raw Training Data* in Fig. 1) are used for the data augmentation step. Four different data augmentation techniques (e.g. *Jittering, Scaling, Rotation*, and *Time-warping*) are implemented to generate synthetic training data. The augmented training data and raw training data are then combined together and used to train the LSTM network. Finally, the test data, which was initially separated from the raw data, are used to evaluate the performance of the trained LSTM network. Two major methodological steps in this study, *Time-series Data Augmentation*, and *Training LSTM Network*, highlighted in *bold* in Fig. 1 are discussed in the following subsections.

4.1. Time-series data augmentation

This section describes data augmentation techniques that can be used to synthetically generate time-series data for deep learning. For image recognition applications, mirroring, scaling, cropping, rotating etc. are legitimate augmentation techniques as minor changes due to these techniques do not alter the label of the image as they may happen in real world observation. However, these label-preserving transformations are not intuitively recognizable for time-series IMU data. Factors that can introduce variability without altering the labels of the time-series data are random noise, sensor placements, and temporal characteristics of activities. In order to account for those factors *Jittering*, *Scaling*, *Rotation*, and *Time-warping* are implemented as shown in Fig. 2 [19]. Fig. 2(a) shows the raw data of one channel of the sensor.

Jittering: Jittering is implemented as a way to simulate additive sensor noise. Each sensor has a different type of mechanical noise. Simulating random sensor noise increases the robustness of the training data against various types of sensors and their multiplicative and additive noises. White Gaussian noise is used in this study to add the jittering to raw training data. The effect of jittering on the test dataset is illustrated in Fig. 2(b).

Scaling: Scaling is another technique adopted in data augmentation which changes the magnitude of the raw data but preserves the labels. This variation observed in situations wherein the dimensions of the implement to which the sensor is attached changes, such as a change in length of excavator boom etc. Scaling is implemented by multiplying the raw training data by a random scalar. Fig. 2(c) shows an augmented dataset after applying scaling to the test data.

Rotation: Rotation can be accounted for introducing label-invariant variability of IMU sensor data when sensors are placed in the equipment with different orientations. For example, an upside-down placement of the sensor can invert the sign of the IMU readings without changing the labels as shown in Fig. 2(d). Moreover, applying arbitrary rotation to the raw data can account for any minor changes in the sensor orientation because of vibration during data collection.

Time-warping: Time-warping is a technique to generate synthetic training data for different temporal characteristics of an equipment for a specific task. For example, a loading activity can be performed by an excavator with various operating speeds. Each activity was warped (i.e. stretched or shortened) with different warping ratios to account for this variability. A sample warped data can be seen in Fig. 2(e).

Each augmentation technique generated 4-fold increase in augmented training data, resulting in a 16-fold increase in the number of the training datapoints. Four different signal-to-noise ratios were used for *Jittering* to generate 4-fold simulated data with different noise levels. For *Scaling*, scalar multiplication values of 0.8, 0.9, 1.1, and 1.2 were selected to generate 4-fold augmented data with slightly different magnitudes of the reading. Similarly, four different rotation factors were selected to change the sign of the IMU reading, which resulted in further 4-fold increase of the training data. Finally, four warping ratios were selected to generate 4-fold synthetic data with various temporal length, preserving their labels.

4.2. Training the LSTM network

An LSTM network is a type of recurrent neural network (RNN) which learns the long-term temporal dependencies between time steps of sequence data. This is particularly effective in this research as IMU data of each activity of the equipment is comprised of a time-series sequence with temporal dependencies. The main components of an LSTM network for time-series classification are a *sequence input layer*, an *LSTM layer*, a *fully connected layer*, a *softmax layer*, and a *classification output layer* as shown in Fig. 3 [48].

Time-series training data were used in *sequence input layer* which inputs the sequences into the network. The *LSTM layer* learns long-term temporal dependencies between time steps of sequence data in terms of weight matrix and bias vector. The *fully connected layer* then multiplies the inputs by the weight matrix and adds the bias vector. Next, the *softmax layer* applies neural transfer function to the input. Finally, the classification output layer computes the cross-entropy loss for multi-class classification problems with mutually exclusive classes. The *LSTM*

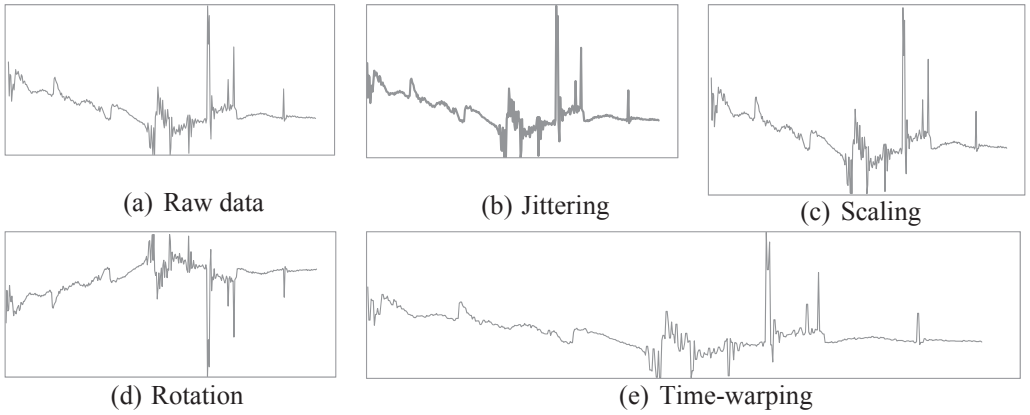

Fig. 2. Time-series data augmentation techniques.

Layer is composed of LSTM units and a common architecture of LSTM units consists of a cell (i.e. *c*), input gate (i.e. *i*), output gate (i.e. *o*), and forget gate (i.e. *f*) as shown in Fig. 4 [48].

Each of these gates computes an activation, using an activation function of the weighted sum. In the figure, i_t, o_t, and f_t represents the activations of input, output, and forget gate respectively at time step *t*, where x_t and h_t are the input and output vector of the LSTM unit. The three exit arrows from the memory cell *c* to 3 gates *i*, *o*, *f* denote the contributions of the activation of the memory cell *c* at time step *t*-1 (i.e. the contribution of c_{t-1}). In other words, the gates *i*, *o*, and *f* calculate their activations at time step *t* considering the activation of the memory cell *c* at time step *t*-1. The circle containing an X symbol in the Fig. 4 represents element-wise multiplication between its inputs. Also the circle containing an S-like curve represents the application of an activation function (e.g. sigmoid function) to a weighted sum [49]. In this study, the input vector *x* contains the time-series sensor readings of the IMUs. Unlikely traditional machine learning approach, where raw data are processes, segmented, and statistical features are extracted, LSTM network can automatically learn high-level representative features containing the long-range temporal relationship between time steps. Thus, raw sensor data and augmented training data are used as input vector in the LSTM to train the deep network. After the training phase, test data are used to evaluate the trained model. The next section discusses the case study used to validate the developed methodology, followed by the results obtained from the model evaluation.

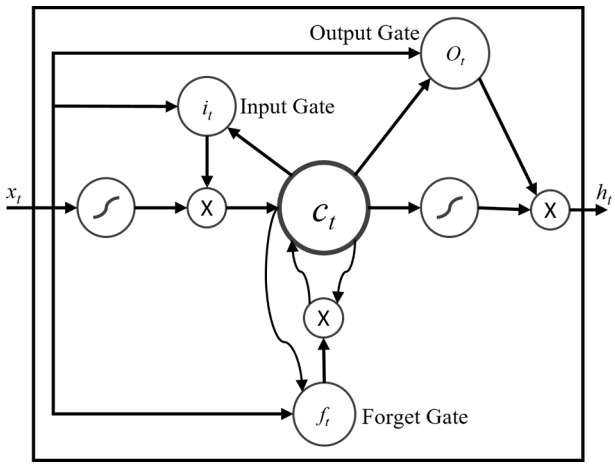

Fig. 4. Architecture of a long-short term memory (LSTM) unit [48].

5. Case study and model evaluation

To evaluate the developed LSTM network, two sets of time-series data were collected from two different types of construction equipment, an excavator and a front-end loader. The collected sensor data were then separated into training and test dataset, where training dataset were used to generate more synthetic training data (i.e. data augmentation), and test data were used to validate the trained model. Several performance matrices were used to evaluate the performance of the LSTM network. Moreover, a comparative analysis was conducted to see the improvement resulted by the developed deep network compared to a traditional shallow network (i.e. ANN).

5.1. Data collection

This study explores the feasibility of using motion data of different articulated implements of the equipment to identify their activities. In the recent years, equipment manufacturers and third-party companies have started mounting motion sensors in the equipment to locate the cutting edge for automated machine guidance (AMG). In the data collection phase, those motion sensors were mimicked using multiple IMUs attached to different elements (e.g. bucket, arm, boom etc.) of the equipment. As opposed to previous studies where smartphone-embedded sensors [2,27] or a single accelerometer [13] was used to capture the vibration of the cabin, this study records the accelerometer and gyroscope data of multiple articulated elements of the equipment.

Two case studies were performed by collecting two sets of motion data from the construction site: from an excavator (Komatsu PC 300 LC), and from a front-end loader (Caterpillar 980G). Both the case studies represent realistic construction operations, i.e., no directions were provided to the equipment operators on how to perform the operations. This was done in order to test the robustness of the developed framework. For each of these equipment, 3 IMU sensors were attached to different articulated parts of the equipment. Fig. 5(a) and (b) shows the placements of the 3 IMU sensors, with white star symbols, on the excavator and the front-end loader respectively, and Fig. 5(c) shows one IMU covered in a plastic box which can be attached to any metal surface with the help of a bottom magnet and further secured with masking tape. Each of the IMUs used in this study can record 3-axis accelerometer, and 3-axis gyroscope data (i.e. total 6 sensor readings per IMU). The three IMU sensors used in this study collected 18 channel of time-series data (i.e. three IMU multiplied by six channel) each equipment. Data was collected for approximately two hours for each

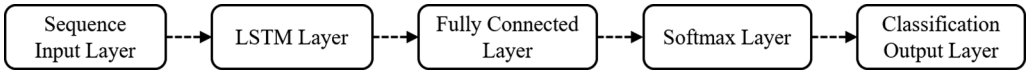

Fig. 3. Main components of an LSTM network.

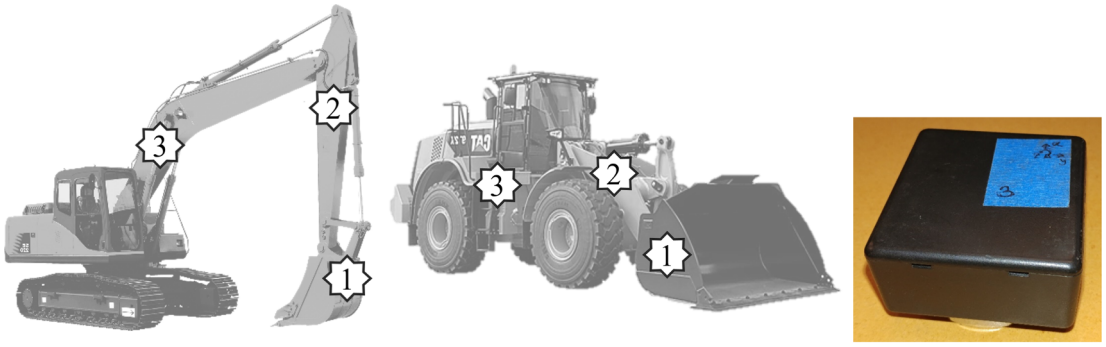

Fig. 5. Placement of the IMU on (a) excavator, (b) front-end loader, and (c) the IMU itself.

equipment. Given an average data capture frequency of 80 Hz, approximately 576,000 data points were collected for each channel of the sensor (e.g. accelerometer X) for each equipment. Furthermore, the equipment was videotaped for the entire duration of data-collection to aid with data labeling and validation purpose of results. Typically, large amount of field data is required to train deep learning network, which may pose a practical challenge in active construction site. This study alleviates this manual field effort by generating large volume of training data from small amount of field data. In practice, generic models for each type of equipment (e.g., excavator, front-end loader, articulated trucks, etc.) can be pre-trained just one time and then implemented in any construction site.

One of the most important aspects of any classification problem is to divide the operation of the equipment into smallest distinguishable activities (i.e. classes) for training and validation. The resolution of the classes (number of separate activities) depends upon the type of equipment, its operation, and the desired application of the analysis [27]. In this study, 9 activities were selected for the excavator, and 10 activities were selected for the front-end loader. Table 1 shows the lists of distinguishable activities for excavator and front-end loader selected in this study. The activities were labelled manually using the reference video captured during the data collection.

The reason for choosing a higher number of classes (compared to previous similar studies such as, [2,13,14]) in this study was to test the robustness of the deep network even when the signal patterns of the IMU of different classes become more similar due to higher resolution. The next subsection discusses the data augmentation techniques implemented using the collected field data.

5.2. Data augmentation

After collecting the field data, activities were categorized and labelled using the videotapes as reference. Instances of each type of activities (e.g., idle, scooping, dumping etc.) were separated into raw training and testing datasets using 50:50 ratio. Typically, in machine learning and deep learning domain, the dataset is separated into training and testing sets using 75:25, 80:20, or 85:25 ratios. In this study, 50% data were separated out (as raw training data) for implementing the augmentation techniques. This was done in corresponding to the overall research objective of minimizing the effort of field data collection. The other 50% were separated as test data, which were used to validate the trained model. Table 2 illustrates the instances (i.e., number of occurrence) of each activity for the excavator in raw data, raw training data, test data, and in each augmented data. For example, there were total 156 scooping activities in the raw data collected from the field for the excavator. 78 (i.e., 50%) were separated for augmentation techniques and 78 (i.e., 50%) for testing the model. Each of the augmentation techniques (e.g., jittering, scaling, rotation, and time-warping) generated 4-fold augmented training data, or 312 number of instances for scooping. Adding the raw training data, and all augmented training data, 1326 instances of scooping activities generated and used for training. Each of the activities for both the excavator and the front-end loader were processed in the same way to generate 16-fold training dataset.

5.3. Implementation of LSTM network

The lengths (i.e., duration) of all the activities were different in the training dataset. Moreover, there were intra-class differences in the activity lengths as well. For example, the average length of *Scooping* activity was 6.3 s, with a range between 2.9 s to 16.9 s, while the average length of *Swing Loaded* activity was 4.9 s, with a range between 1.9 s to 12.9 s. Unlike the classification algorithms, where the window size is required to be constant, LSTM network can take inputs of different lengths and treat each data point as a separate input. However, this approach takes significant computational time. As this study compares effects of different data augmentation techniques and their combinations, several LSTM networks needed to be trained. Thus, in order to make the training process less time consuming, and computationally efficient, each labelled activity sequence was segmented with 1 s (i.e., 76 data points) window size. Each window was represented with the same label as the activity it was segmented from. There were 18 features (i.e., three IMUs X 6 data stream per IMU) in each window. Thus, the dimension of one window was 18 X 76. No statistical feature was extracted from the windows i.e. raw accelerometer and gyroscope data were used as feature vectors. These windows and their corresponding labels were used as inputs, and outputs for the training the LSTM networks. Bidirectional LSTM was used with 100 hidden units. 20 epochs, 850 iteration per epoch, and learning rate of 0.001 was used to train the model. Matlab 2018b in a desktop computer [Intel Xeon CPU @ 3.40 GHz, 16 GB RAM, Windows 10, 64-bit] was used for the training and evaluation of the LSTM networks.

Table 1
Name of the selected activities for the excavator and the front-end loader.

Equipment type	Name of the activities
Excavator	*Engine off, idle, scoop, dump, swing loaded, swing empty, move forward, move backward, and level ground*
Front-end loader	*Engine off, idle, scoop, raise, dump, lower, move forward loaded, move backward loaded, move forward empty, move backward empty*

Table 2
Number of instances of each activity before and after data augmentation.

Activity name	Collected data	Raw training data	Test data	Jittering (4-fold)	Scaling (4-fold)	Rotation (4-fold)	Time-warping (4-fold)	Total training data (16-fold)
Engine off	322	166	166	664	664	664	664	2822
Idle	68	34	34	136	136	136	136	578
Scooping	156	78	78	312	312	312	312	1326
Dumping	162	81	81	324	324	324	324	1377
Swing Loaded	160	80	80	320	320	320	320	1360
Swing Empty	174	87	87	348	348	348	348	1479
Moving Forward	10	5	5	20	20	20	20	85
Moving Backward	14	7	7	28	28	28	28	119
Leveling	34	17	17	68	68	68	68	289
Total	1110	555	555					9435

5.4. Performance measures

In this study, four common performance measures; accuracy, precision, recall, and F_1 score were used to measure the performance of the LSTM network and to compare the LSTM network with ANN. Accuracy of the classification model can be calculated by dividing the number of correctly classified class with total number of classes as shown in Eq. (1).

$$Accuracy = \frac{Number\ \ of\ \ correctly\ \ classified\ \ classes}{Total\ \ number\ \ of\ \ classes} \times 100\% \quad (1)$$

Precision and recall of the model are calculated to account the cost associated with misclassification. Precision is the fraction of predicted positive instances (i.e. true positive + false positive) that are truly positive (i.e. true positive), while recall refers to the fraction of true instances (i.e. true positive + false negative) that are correctly predicted as positive (i.e. true positive). Precision and recall can be mathematically expressed by Eqs. (2) and (3).

$$Precision = \frac{True\ \ Positive}{True\ \ Positive + False\ \ Positive} \times 100\% \quad (2)$$

$$Recall = \frac{True\ \ Positive}{True\ \ Positive + False\ \ Negative} \times 100\% \quad (3)$$

While it is desirable to achieve high precision and recall value, it is often challenging to maximize both measures for a single classification model. Thus, F_1 score is calculate which is the harmonic mean of precision and recall, as shown in Eq. (4).

$$F_1 Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (4)$$

The following section discusses the evaluation of the deep network using performance matrices.

6. Results

From a model evaluation perspective, this study focuses on the following three questions:

1. Does deep learning outperform shallow learning?
2. Do data augmentation techniques improve the performance of the deep learning model?
3. Does data augmentation help to reduce the inter-class confusions of the LSTM network?

This section is organized by first summarizing all the performance measures in tabular forms. Then comparative performance analysis of ANN and LSTM network is conducted. The impacts of data augmentation techniques on the performance measures of both ANN and LSTM are investigated. Finally, a closer look at the confusion matrices of LSTM network with and without data augmentation explores the inter-class confusion of the model.

6.1. Summarizing the performance measures

As discussed in Section 4.1, each of the augmentation technique (e.g. jittering, scaling etc.) were implemented using four different technical parameters. The volume of augmented training data is highest when each of the technique was implemented four times, and lowest when each of the technique was implemented just one time. In Section 4.1, while each of the augmentation techniques were applied four times, it was mentioned to have 16- fold augmentation. However, for easy understanding, the following sections in this paper mention 4-fold augmentation when each of the technique was implemented 4 times (i.e. total 16-fold data increase). Similarly, if each of the technique was applied 2 times, it is mentioned to have 2-fold augmentation (i.e., total 8-fold data increase). 5 different volumes of training dataset (i.e. no augmentation, 1-fold, 2-fold, 3-fold, and 4-fold augmentation), as shown in Table 2, were generated and the LSTM network was trained with each of them to evaluate the impact of data augmentation on the performance of the model. Moreover, a shallow network, ANN was also trained for each of the training dataset to compare it with the LSTM network. This should be noted that the test data were separated before the data augmentation and used to evaluate the LSTM network and the ANN. This helps to compare the models and to evaluate the impact of data augmentation. Tables 3 and 4 summarizes all the performance measures for both ANN and LSTM network trained with the training data and tested with test data for both the excavator and the loader. No detailed discussion is provided on the training accuracy as the models were evaluated using the test data, separated from the raw data at the beginning. However, the overall test accuracy ranged between 85% and 99% for all the trained models. The left most column of Tables 3 and 4 shows the amount of data augmentation. *No Aug.* represents only the raw training data (i.e. no augmentation), and *4-fold* represents augmenting raw training data 4 times with each of the augmentation technique. Each of the performance measure is listed side-by-side in the tables for an easy comparison of ANN and LSTM. These two tables are deconstructed with different types of data visualization (e.g. bar chart, line graph etc.) to address the three aforementioned questions of this section.

Table 3
Performance measures of ANN and LSTM for the **excavator**.

	Accuracy		Precision		Recall		F-1 Score	
	ANN	LSTM	ANN	LSTM	ANN	LSTM	ANN	LSTM
No Aug.	62.2%	63.3%	50.5%	55.1%	54.0%	54.1%	51.3%	54.1%
1-fold	74.9%	94.0%	63.9%	91.3%	71.8%	92.7%	65.8%	91.9%
2-fold	78.1%	97.1%	69.9%	92.9%	73.3%	93.9%	70.9%	93.3%
3-fold	78.7%	97.9%	70.3%	95.9%	73.5%	97.8%	71.3%	96.7%
4-fold	79.9%	97.9%	70.7%	96.2%	74.8%	99.0%	71.3%	97.6%

Table 4
Performance measures of ANN and LSTM for the **loader**.

	Accuracy		Precision		Recall		F-1 Score	
	ANN	LSTM	ANN	LSTM	ANN	LSTM	ANN	LSTM
No Aug.	48.8%	59.7%	36.6%	52.6%	47.1%	54.7%	35.4%	52.4%
1-fold	62. 6%	78.7%	51.1%	75.8%	61.9%	77.9%	51.8%	76.6%
2-fold	63.8%	94.1%	52.7%	93.3%	61.6%	93.1%	53.7%	93.2%
3-fold	64.1%	95.4%	54.2%	93.7%	63.1%	95.1%	55.5%	94.4%
4-fold	66.1%	96.7%	56.1%	96.3%	64.7%	96.4%	57.1%	96.3%

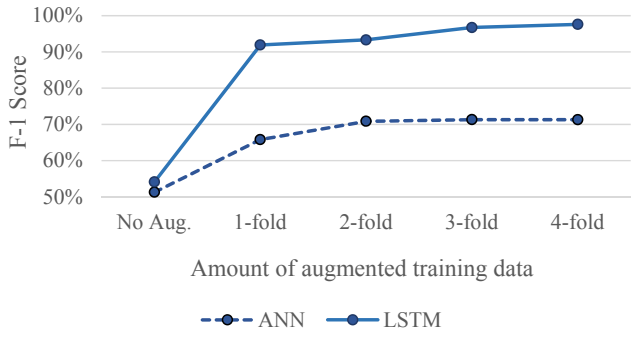

Fig. 6. F-1 score with different amount of augmented training data for the **excavator**.

6.2. Performance of ANN vs. LSTM network

Figs. 6 and 7 plots the F-1 score for both ANN and LSTM network for the excavator, and the loader respectively. The x-axis shows the amount of augmented training data.

From Figs. 6 and 7, it is evident that the LSTM performance increases with increase in training data at a higher rate than ANN. For example, the F-1 score of LSTM and ANN are 54.1% and 51.3% (i.e. difference of 2.8%) respectively for the excavator when trained with only raw training data (i.e. no augmentation). However, when the amount of training data is increased to 1-fold, the difference in F-1 score is 26.1% (i.e. 91.9% for LSTM and 65.8% for ANN). The similar characteristic is seen in Fig. 7 as well. This exponential increase in performance supports the argument that deep networks outperform shallow networks with larger training dataset [50].

6.3. Impact of volume of training data

The performance measures for both ANN and LSTM are visualized in Fig. 8 (excavator) and Fig. 9 (loader).

From both the figures, a positive impact of data augmentation on the each of performance measures can be observed. Accuracy, precision, recall and F-1 score increases with each phase (e.g. 1-fold, 2-fold etc.) of data augmentation. Specifically, significant improvement is noticed from *No Aug.* to *1-fold* augmentation. For example, precisions of the LSTM network for the excavator are 55.1%, 91.3%, 92.9%, 95.9%, and 96.2% for *No Aug.*, *1-fold*, *2-fold*, *3-fold*, and *4-fold* augmentation. This supports the argument that data augmentation techniques can generate synthetic training data by infusing variations to the raw data without altering their classes. We see an overall improvement of 34.6% in accuracy, 41.1% in precision, 44.9% in recall, and 43.5% of F-1 score for the LSTM network after applying the data augmentation for excavator. Similarly, LSTM network for the loader illustrates significant improvement of the performance measures after introducing data augmentation.

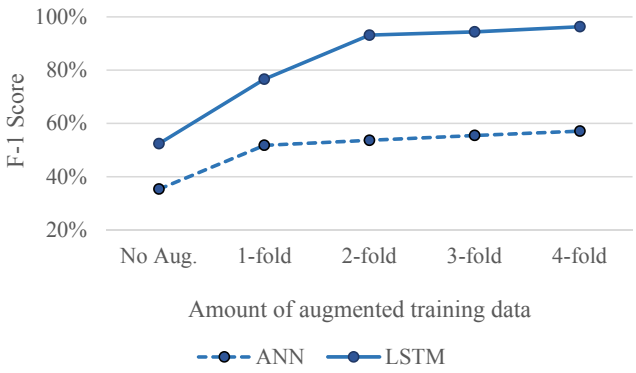

Fig. 7. F-1 score with different amount of augmented training data for the **loader**.

6.4. Impact of augmentation types

This section discusses the result of a sensitivity analysis to find the impact of different augmentation techniques on the performance of LSTM network. Fig. 10 summarizes all four performance measures (i.e., accuracy, precision, recall, and F1 score) for different augmentation techniques. From this figure, this is evident that, *Jittering, Scaling, and Time-warping* have positive impact on the performance of LSTM. On the other hand, applying *Rotation* have little or no impact on the LSTM network.

6.5. Detailed comparison using confusion matrix

Even though accuracy, precision, recall, and F-1 score represent overall performance of the LSTM network, they do not provide any information on how instances are misclassified. Thus, confusion matrices are introduced to identify the classes that are misclassified and confused with other classes. Figs. 11 and 12 shows the confusion matrices with and without data augmentation for the excavator and the loader respectively. Each row represents the actual classes, and columns represent predicted classes. The green diagonal cells in these figures represent correctly classified classes, where all other cells show the misclassified classes.

From Fig. 11, we see that the top two misclassified activities for the excavator are *Dump* and *Level Ground* before data augmentation. *Dump* is confused 236 times with the *Level Ground* (highlighted by black border), where *Level Ground* is confused 140 times with the *Scoop* (highlighted by red border). During the *Level Ground* activity, the excavator was picking up a small amount of soil (similar to *Scoop*) and dumping them in close proximity (similar to *Dump*). Thus, the confusion among *Level Ground, Dump, and Scoop* can be explained by the similar signal patterns generated from the IMUs. However, after applying the augmentation these confusions are noticed to be reduced 236 to 2 times for the *Dump* activity and 140 to 4 times for the *Level Ground* activity. We see a similar situation in Fig. 12, where 252 instances of confusion between *Lower* and *Move For. Empty* is reduced 7 instances, and 290 instances of confusion between *Move Bac. Empty* and *Move For. Empty* is reduced to 12 instances. Significant reduction of misclassified instances is noticeable in the figures after applying data augmentation. This supports the argument that introducing slight variations (i.e. data augmentation) to the raw training data to generate more data helps better generalization of the deep network.

7. Discussion

The proposed framework in this paper consists of two major components: a LSTM-based deep learning activity recognition framework for construction equipment, and time-series data augmentation techniques to generate synthetic training data. The deep learning network outperformed the shallow network in terms of accuracy, precision, recall, and F-1 score. It was also observed that data augmentation techniques enriched the volume of training dataset without altering their

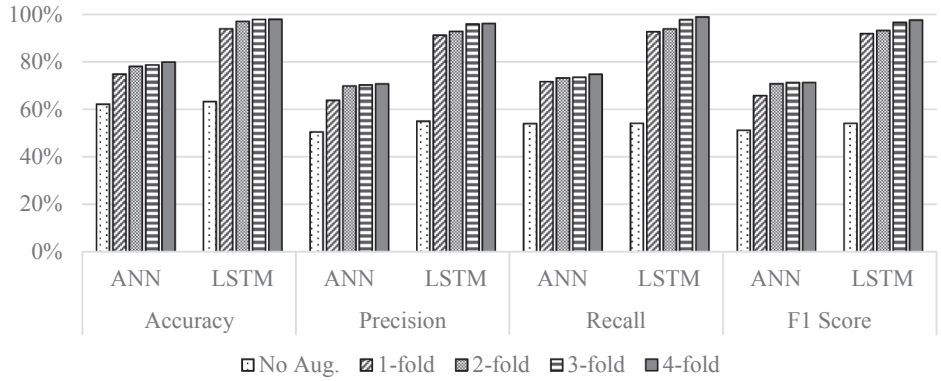

Fig. 8. Performance measures of ANN and LSTM for the **excavator**.

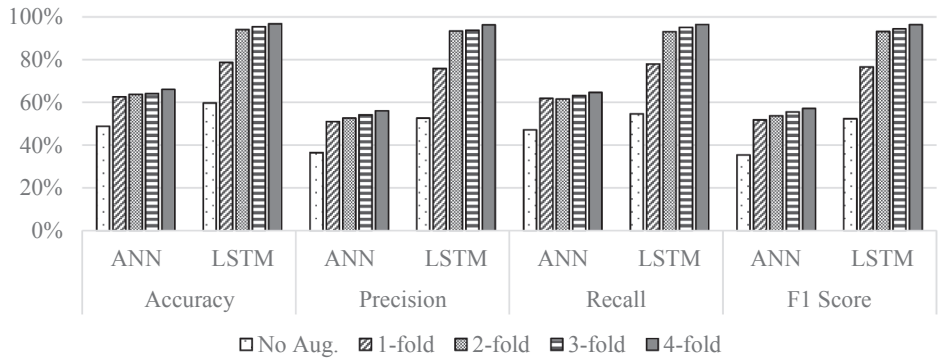

Fig. 9. Performance measures of ANN and LSTM for the **loader**.

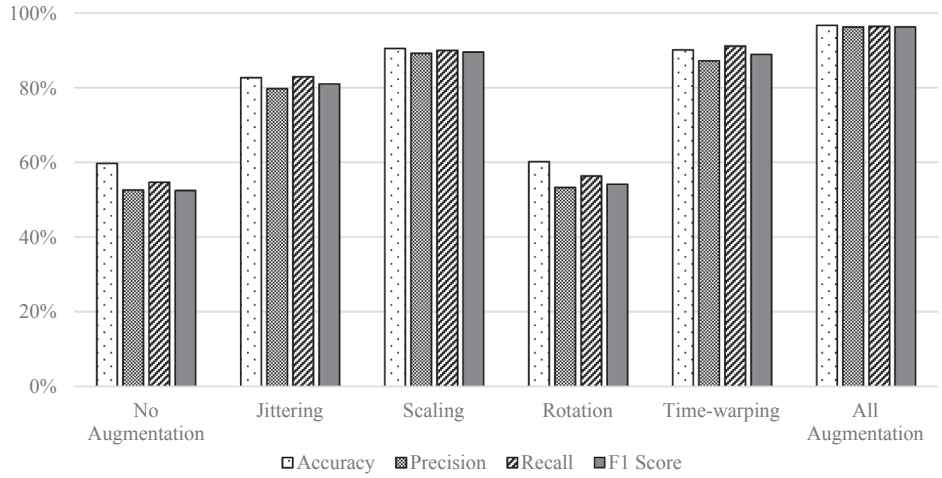

Fig. 10. Impact of different data augmentation techniques on the performance of LSTM network.

predefined labels. This section explicitly summarizes the contributions of this research that be inferred from both of those components of the research.

The major contributions of the use of LSTM network in this study are:

- The LSTM network eliminated the necessity of manual feature extraction, which is limited by human knowledge of the domain. Instead, the deep network automatically learned high-level representative features from the raw training data.
- As opposed to the traditional classification algorithms (e.g., ANN, KNN, SVM etc.), the LSTM network contains long-term temporal dependency of the training data between consecutive time steps.

The major contributions of the data augmentation components in this study are:

- Implementation of data augmentation eliminated the necessity of collecting large volume of training data from the construction site. This improves the practicality of such classification techniques for temporary and transient construction operations.
- Synthetic training data removed bias in the trained model due to

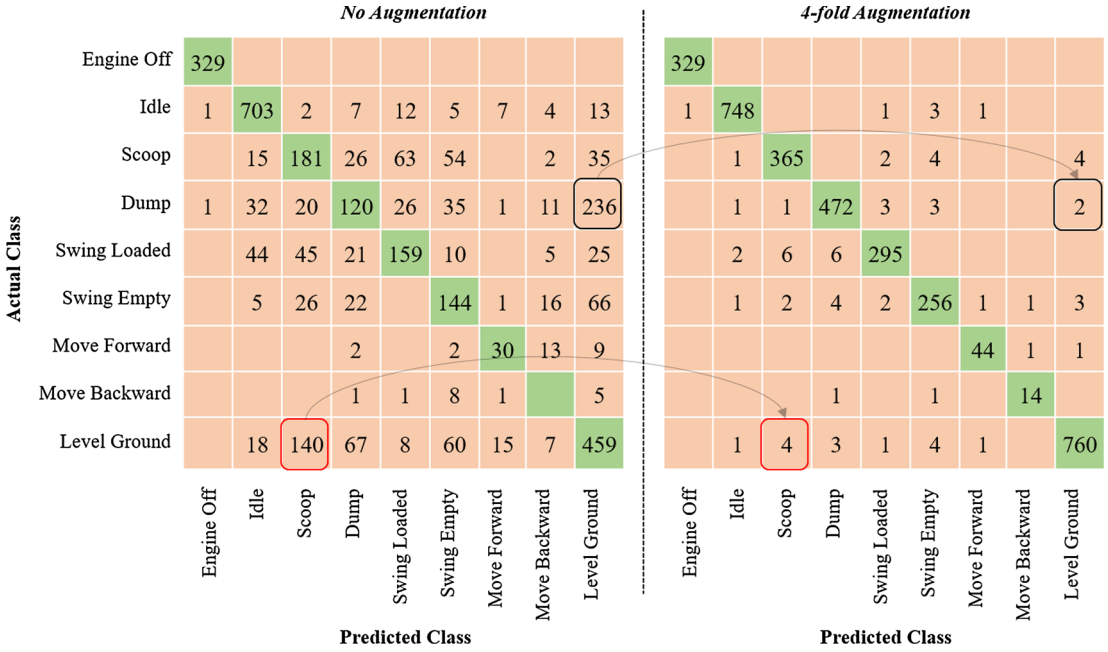

Fig. 11. Confusion matrix of LSTM network for the **excavator**.

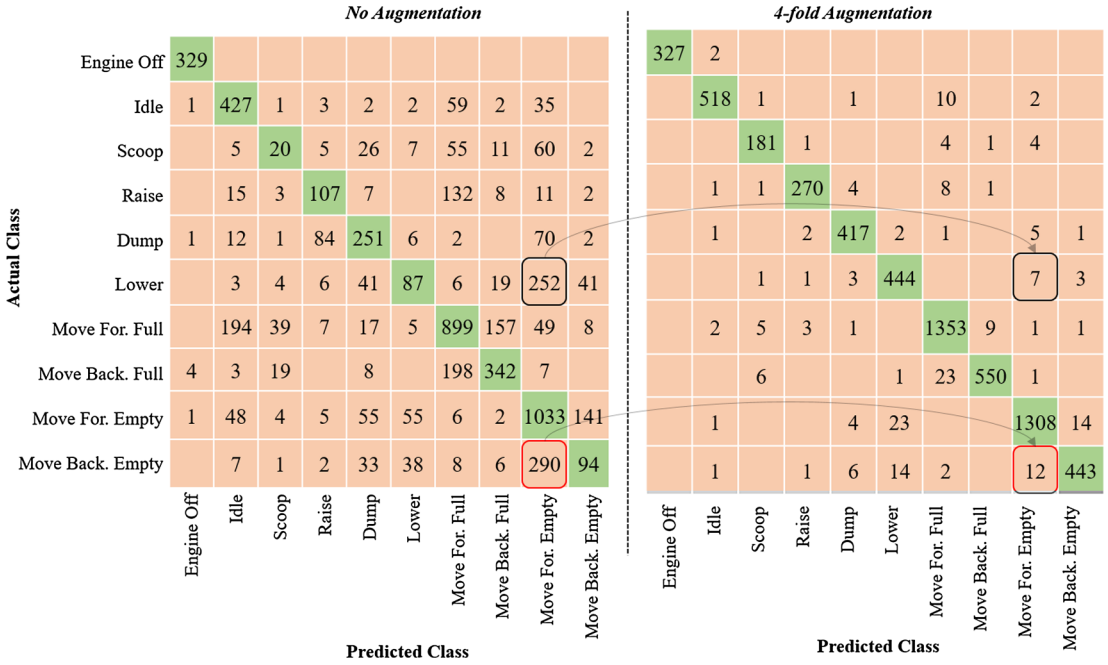

Fig. 12. Confusion matrix of LSTM network for the **loader**.

imbalanced volume of training data. For example, if there is little training data of one specific class compared to other classes, the trained model might be biased towards that class.

8. Conclusions and future work

Automated, real-time, and reliable activity recognition framework for construction equipment provides a foundational platform to monitor and assess productivity, safety, and environmental impact on construction site. Towards this end, this paper provides an LSTM-based activity recognition framework for construction equipment using multiple IMUs attached to different articulated elements of the equipment. Moreover, time-series data augmentation techniques were developed and implemented to generate synthetic training data, reducing the necessity of large volume of data collection from construction site. The developed framework is validated using two case studies using an excavator and a front-end loader. The results of the case studies show that the deep learning approach (i.e. LSTM network) outperforms the shallow network (i.e. ANN). The data augmentation techniques developed and implemented in this research shows the ability to correctly simulate real-world training dataset. This helps rigorous training of deep network without collecting large amount of training data from the field. Moreover, implementation of data augmentation helps to reduce inter-class confusion of the LSTM network. Successful application of the proposed framework has the potential to transform the way construction operations are currently being monitored. By executing the proposed framework in real construction site, construction operations can be continuously monitored and assessed in real-time, relieving

construction companies from time-consuming and subjective manual method of analyzing construction operations.

The proposed framework is capable of identifying activities of two specific construction equipment (e.g. excavator, and front-end loader). The designed methodology will be broadened to cover other type of equipment to ensure its robustness. Training LSTM requires significantly higher computational power and time than training a shallow network. However, in practice, models can be trained for specific equipment just once and then used repeatedly. Another challenge in this study was the manual labeling of the data, which takes significant effort and time. Future works of this study include using the activity level information for productivity analysis, safety analysis, and fuel use analysis techniques to support better decision-making and control methods. In addition to that, the proposed framework will be extended for human workers in construction site using wearables to enable productivity and safety applications.

Declaration of Competing Interest

None.

References

[1] 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.

[2] N. Mathur, S.S. Aria, T. Adams, C.R. Ahn, S. Lee, Automated cycle time measurement and analysis of excavator's loading operation using smart phone-embedded IMU sensors, in: International Workshop on Computing in Civil Engineering, Austin, TX, 2015, pp. 215–222. doi:10.1061/9780784479247.027.

[3] H. Kim, C.R. Ahn, D. Engelhaupt, S. Lee, Application of dynamic time warping to the recognition of mixed equipment activities in cycle time measurement, Autom. Constr. 87 (2018) 225–234, https://doi.org/10.1016/j.autcon.2017.12.014.

[4] F. Vahdatikhaki, A. Hammad, Framework for near real-time simulation of earthmoving projects using location tracking technologies, Autom. Constr. 42 (2014) 50–67, https://doi.org/10.1016/j.autcon.2014.02.018.

[5] S.C. Ok, S.K. Sinha, Construction equipment productivity estimation using artificial neural network model, Constr. Manage. Econ. (2006), https://doi.org/10.1080/01446190600851033.

[6] J. Gong, C.H. Caldas, An intelligent video computing method for automated productivity analysis of cyclic construction operations, in: Proceedings of the ASCE International Workshop on Computing in Civil Engineering, 2009, pp. 64–73. doi:10.1061/41052(346)7.

[7] J. Zou, H. Kim, Using hue, saturation, and value color space for hydraulic excavator idle time analysis, J. Comput. Civil Eng. 21 (2007) 238–246, https://doi.org/10.1061/(ASCE)0887-3801(2007) 21:4(238).

[8] J. Seo, S. Han, S. Lee, H. Kim, Computer vision techniques for construction safety and health monitoring, Adv. Eng. Inf. (2015), https://doi.org/10.1016/j.aei.2015.02.001.

[9] T. Cheng, J. Teizer, Real-time resource location data collection and visualization technology for construction safety and activity monitoring applications, Autom. Constr. 34 (2013) 3–15, https://doi.org/10.1016/j.autcon.2012.10.017.

[10] K.M. Rashid, A.H. Behzadan, Risk behavior-based trajectory prediction for construction site safety monitoring, J. Constr. Eng. Manage. 144 (2018) 04017106, https://doi.org/10.1061/(ASCE)CO.1943-7862.0001420.

[11] K.M. Rashid, S. Datta, H. Behzadan, Amir, Coupling risk attitude and motion data mining in a preemtive construction safety framework, in: Proceeding of Winter Simulation Conference, IEEE, Las Vegas, NV, 2017, pp. 4220–4227.

[12] A. Martín-Garín, J.A. Millán-García, A. Baïri, J. Millán-Medel, J.M. Sala-Lizarraga, Environmental monitoring system based on an open source platform and the internet of things for a building energy retrofit, Autom. Constr. 87 (2018) 201–214, https://doi.org/10.1016/j.autcon.2017.12.017.

[13] 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. Civil Eng. 29 (2015) 04014042, https://doi.org/10.1061/(ASCE)CP.1943-5487.0000337.

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

[15] J. Louis, P.S. Dunston, Methodology for real-time monitoring of construction operations using finite state machines and discrete-event operation models, J. Constr. Eng. Manage. 143 (2017) 04016106, https://doi.org/10.1061/(ASCE)CO.1943-7862.0001243.

[16] A.H. Behzadan, V.R. Kamat, A framework for utilizing context-aware augmented reality visualization in engineering education, in: International Conference on Construction Application of Virtual Reality, 2012, pp. 292–299.

[17] S. Dong, V.R. Kamat, SMART: scalable and modular augmented reality template for rapid development of engineering visualization applications, Visualization Eng. 1 (2013) 1–17, https://doi.org/10.1186/2213-7459-1-1.

[18] J. Louis, P. Dunston, Platform for Real Time Operational Overview of Construction Operations, Construction Research Congress ASCE, 2016, pp. 2039–2049. doi:10.1061/9780784479827.203.

[19] T.T. Um, F.M.J. Pfister, D. Pichler, S. Endo, M. Lang, S. Hirche, U. Fietzek, D. Kulić, Data augmentation of wearable sensor data for Parkinson's disease monitoring using convolutional, Neural Networks (2017), https://doi.org/10.1145/3136755.3136817.

[20] E. Rezazadeh-Azar, B. McCabe, Part based model and spatial-temporal reasoning to recognize hydraulic excavators in construction images and videos, Automation Constr. 24 (2012) 194–202, https://doi.org/10.1016/j.autcon.2012.03.003.

[21] R. Bao, M.A. Sadeghi, M. Golparvar-Fard, Characterizing construction equipment activities in long video sequences of earthmoving operations via kinematic features, in: Construction Research Congress, ASCE, San Juan, Puerto Rico, 2016, pp. 849–858. doi:10.1061/9780784479827.203.

[22] J. Kim, S. Chi, J. Seo, Interaction analysis for vision-based activity identification of earthmoving excavators and dump trucks, Autom. Constr. 87 (2018) 297–308, https://doi.org/10.1016/j.autcon.2017.12.016.

[23] J. Gong, C.H. Caldas, C. Gordon, Learning and classifying actions of construction workers and equipment using Bag-of-Video-Feature-Words and Bayesian network models, Adv. Eng. Inf. 25 (2011) 771–782, https://doi.org/10.1016/j.aei.2011.06.002.

[24] S. El-Omari, O. Moselhi, Integrating automated data acquisition technologies for progress reporting of construction projects, Autom. Constr. 20 (2011) 699–705, https://doi.org/10.1016/j.autcon.2010.12.001.

[25] E. Ergen, B. Akinci, B. East, J. Kirby, Tracking components and maintenance history within a facility utilizing radio frequency identification technology, J. Comput. Civil Eng. 21 (2007) 11–20, https://doi.org/10.1061/(ASCE)0887-3801(2007)21:1(11).

[26] L. Song, N.N. Eldin, Adaptive real-time tracking and simulation of heavy construction operations for look-ahead scheduling, Autom. Constr. 27 (2012) 32–39, https://doi.org/10.1016/j.autcon.2012.05.007.

[27] R. Akhavian, L. Brito, A. Behzadan, Integrated mobile sensor-based activity recognition of construction equipment and human crews, in: Conference on Autonomous and Robotic Construction of Infrastructure, Iowa State University, Ames, Iowa, 2015, pp. 1–20.

[28] P. Vepakomma, D. De, S.K. Das, S. Bhansali, A-Wristocracy: Deep learning on wrist-worn sensing for recognition of user complex activities, in: 2015 IEEE 12th International Conference on Wearable and Implantable Body Sensor Networks, BSN 2015, 2015, 1–6. doi:10.1109/BSN.2015.7299406.

[29] K. Walse, R.V. Dharaskar, PCA-based optimal ANN classifiers for human activity recognition using mobile sensors data, 50 (2016). doi:10.1007/978-3-319-30933-0.

[30] N.Y. Hammerla, S. Halloran, T. Plötz, Deep, convolutional, and recurrent models for human activity recognition using wearables, IJCAI International Joint Conference on Artificial Intelligence. 2016–Janua (2016) 1533–1540.

[31] Y. Bengio, Deep learning of representations: Looking forward, Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). 7978 LNAI (2013) 1–37. doi:10.1007/978-3-642-39593-2_1.

[32] M. Zeng, L.T. Nguyen, B. Yu, O.J. Mengshoel, J. Zhu, P. Wu, J. Zhang, Convolutional neural networks for human activity recognition using mobile sensors, in: Proceedings of the 6th International Conference on Mobile Computing, Applications and Services. 6 (2014). doi:10.4108/icst.mobicase.2014.257786.

[33] J.B. Yang, M.N. Nguyen, P.P. San, X.L. Li, S. Krishnaswamy, Deep convolutional neural networks on multichannel time series for human activity recognition, IJCAI International Joint Conference on Artificial Intelligence. 2015–Janua (2015) 3995–4001. doi:10.3897/zookeys.77.769.

[34] M. Inoue, S. Inoue, T. Nishida, Deep recurrent neural network for mobile human activity recognition with high throughput, Artificial Life Robotics 23 (2016) 173–185, https://doi.org/10.1007/s10015-017-0422-x.

[35] F.J. Ordóñez, D. Roggen, Deep convolutional and LSTM recurrent neural networks for multimodal wearable activity recognition, Sensors (Switzerland) 16 (2016), https://doi.org/10.3390/s16010115.

[36] A. D'Innocente, F.M. Carlucci, M. Colosi, B. Caputo, Bridging between computer and robot vision through data augmentation: A case study on object recognition, lecture notes in computer science (including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics), 10528 LNCS (2017) 384–393, https://doi.org/10.1007/978-3-319-68345-4_34.

[37] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition kaiming, 2016, pp. 1–9. doi:10.3389/fpsyg.2013.00124.

[38] C.C. Charalambous, A.A. Bharath, A data augmentation methodology for training machine/deep learning gait recognition algorithms, 2016, pp. 1–12. doi:10.5244/C.30.110.

[39] A. Radford, L. Metz, S. Chintala, Unsupervised representation learning with deep convolutional generative adversarial networks, 2015, pp. 1–16. doi:10.1051/0004-6361/201527329.

[40] M. Liang, X. Hu, Recurrent convolutional neural network for object recognition, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 07–12–June, 2015, pp. 3367–3375. doi:10.1109/CVPR.2015.7298958.

[41] R.T. and Y.O. Naoyuki Kanda, Elastic Spectral Distortion for Low Resource Research Laboratory , Hitachi Ltd ., 2013, pp. 309–314.

[42] N. Jaitly, G.E. Hinton, Vocal tract length perturbation (VTLP) improves speech recognition, in: Proc. ICML Workshop on Deep Learning for Audio, Speech and Language, 2013.

[43] J. Schlüter, T. Grill, Exploring data augmentation for improved singing voice detection with neural networks, 2013. https://grrrr.org/pub/schlueter-2015-ismir.

pdf.
[44] A. Le Guennec, S. Malinowski, R. Tavenard, Data augmentation for time series classification using convolutional neural networks, 2016. https://halshs.archives-ouvertes.fr/halshs-01357973 (accessed October 15, 2018).
[45] G. Forestier, F. Petitjean, H.A. Dau, G.I. Webb, E. Keogh, Generating synthetic time series to augment sparse datasets, Proceedings - IEEE International Conference on Data Mining, ICDM. 2017–Novem, 2017, pp. 865–870. doi:10.1109/ICDM.2017.106.
[46] J. Ding, B. Chen, H. Liu, M. Huang, Convolutional neural network with data augmentation for SAR target recognition, IEEE Geosci. Remote Sens. Lett. 13 (2016) 364–368, https://doi.org/10.1109/LGRS.2015.2513754.
[47] A. Le Guennec, S. Malinowski, R. Tavenard, Data augmentation for time series classification using convolutional neural networks, 2nd ECML/PKDD Workshop on Advanced Analytics and Learning on Temporal Data, 2016. https://aaltd16.irisa.fr/files/2016/08/AALTD16_paper_9.pdf.
[48] A. Graves, A.R. Mohamed, G. Hinton, Speech recognition with deep recurrent neural networks, ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing – Proceedings, 2013, pp. 6645–6649. doi:10.1109/ICASSP.2013.6638947.
[49] K. Greff, R.K. Srivastava, J. Koutn\'\ik, B.R. Steunebrink, J. Schmidhuber, LSTM: search space odyssey, CoRR (2015) 2222–2232, https://doi.org/10.1109/TNNLS.2016.2582924.
[50] A. Schindler, T. Lidy, A. Rauber, Comparing shallow versus deep neural network architectures for automatic music genre classification, (n.d.). http://ceur-ws.org.