Recurrent neural networks with long term temporal dependencies in machine tool wear diagnosis and prognosis

Jianlei Zhang¹ · Yukun Zeng² · Binil Starly¹

Received: 4 November 2020 / Accepted: 24 February 2021 / Published online: 9 March 2021
© The Author(s) 2021

Abstract

Data-driven approaches for machine tool wear diagnosis and prognosis are gaining attention in the past few years. The goal of our study is to advance the adaptability, flexibility, prediction performance, and prediction horizon for online monitoring and prediction. This paper proposes the use of a recent deep learning method, based on Gated Recurrent Neural Network architecture, including Long Short Term Memory (LSTM), which try to captures long-term dependencies than regular Recurrent Neural Network method for modeling sequential data, and also the mechanism to realize the online diagnosis and prognosis and remaining useful life (RUL) prediction with indirect measurement collected during the manufacturing process. Existing models are usually tool-specific and can hardly be generalized to other scenarios such as for different tools or operating environments. Different from current methods, the proposed model requires no prior knowledge about the system and thus can be generalized to different scenarios and machine tools. With inherent memory units, the proposed model can also capture long-term dependencies while learning from sequential data such as those collected by condition monitoring sensors, which means it can be accommodated to machine tools with varying life and increase the prediction performance. To prove the validity of the proposed approach, we conducted multiple experiments on a milling machine cutting tool and applied the model for online diagnosis and RUL prediction. Without loss of generality, we incorporate a system transition function and system observation function into the neural net and trained it with signal data from a minimally intrusive vibration sensor. The experiment results showed that our LSTM-based model achieved the best overall accuracy among other methods, with a minimal Mean Square Error (MSE) for tool wear prediction and RUL prediction respectively.

Keywords Recurrent neural networks · Long term temporal dependencies · Long short term memory · Tool wear · Remaining useful life

1 Introduction

DATA-DRIVEN approaches to predicting machine conditions have seen rapid advancements due to low-cost sensor technology and reduced computational cost. In machining based operations, a critical asset to obtaining superior part quality is the cutting tool inserts. These inserts continuously wear out and are replaced in a production based

environment. The continuous estimation of the tool wear classifications caused by abrasion, erosion or tool breakage is necessary to improve the reliability of the metal-cutting manufacturing processes. Tool breakage can be detected by robust power monitoring techniques, leading to emergency shut-off of the machine. However, tool wear due to abrasion and erosion is hard to estimate in continuous production environments. At present, production

✉ Jianlei Zhang, jzhang45@ncsu.edu; Yukun Zeng, yzeng@tamu.edu; Binil Starly, bstarly@ncsu.edu | ¹Department of Industrial and Systems Engineering, North Carolina State University, Raleigh, USA. ²Department of Computer Science and Engineering, Texas A&M University, College Station, USA.

environments resort to offline detection through the use of in-machine laser probes which measure wear amounts on the cutting tool surface or a scheduled replacement of cutting tools based on regular machining time intervals. Non-optimized changeover of cutting tool inserts can lead to lost production efficiency and reduced part quality. Detection and prediction of tool wear amounts in real-time are necessary to improve discrete manufacturing systems operations.

Research in the diagnosis and prediction of tool wear has been carried out for decades. Many state-of-the-art condition-based maintenance (CBM) approaches have been proposed for machinery diagnosis and prognosis. Typically, those methods sought to recommend maintenance decisions based on comprehensive analysis of event and monitoring data from built-in information systems or various sensors installed on the target machine [1]. Event data usually includes discrete events happening on the target machine based on manual entries, e.g., manual installation, scheduled maintenance, abrupt breakdown, etc. Monitoring data, on the other hand, are usually collected continuously from various sensors to track machine conditions such as for vibration and acoustic information. In this paper, we consider the RUL prediction problem with a focus on monitoring data from vibration sensors as the first step. We expect to incorporate the event data in our future models as they can provide invaluable information in making accurate predictions. The RUL prediction problem is particularly difficult due to modeling difficulties involved in a non-linear time-variant process. To cope with this problem, prediction techniques have been applied to estimate future tool wear amounts but often resort to intrusive sensing techniques which are limited in practical industrial implementation. Sensor signals are dependent on a number of factors including machine state, machining conditions and position of the sensor. Adding to the complexity is the apparent variability in the cutting tool, inconsistencies in the material properties of the workpiece and the larger number of machining conditions that must be accounted for.

Several methods have been proposed in the literature for tool wear estimation and prognosis through both linear and non-linear prediction. The most widely used non-linear prediction for tool wear methods involves neural networks (NN). Various works have shown the power of NN in terms of predicting tool wear amounts that can meet accuracy requirements. They do however require large data to collect over periods of time under various machining conditions. Recurrent Neural Networks (RNN) were proposed as an improvement over regular feed-forward neural networks to account for cyclic connections over time. Each successive time-step in the machining states is stored in the internal state of the network to provide a

temporal memory. However, conventional RNN did suffer from gradient vanishing or exploding issues [2] when it is trained with gradient-based learning and back-propagation through time (BPTT) [3] methods. New variations in RNN, the gated recurrent neural networks, including long short-term memory (LSTM) and Gated Recurrent Units (GRU), have been developed to address this problem. Long Short-Term Memory (LSTM) is an RNN architecture that is excellent at remembering both short-term and long-term dependencies. Originally designed by Hochreiter et al. [4], this method was developed to address the issue of the vanishing and/or exploding gradient problem when the number of layers in the neural network is increased. With the introduction of special units with forget gate and cell memory into normal RNNs, LSTM captures complex and artificially long time lag information. Consequently, RNNs in the form of LSTMs have been very successfully used in long, sequential learning tasks such as speech recognition, language modeling, network traffic regulation, visual recognition and many others. In this paper, we make the key observation that accurate RUL prediction relies on a long-term memory on historical tool wear conditions. While mostly omitted in existing methods [5, 6], temporal dependencies have been studied and modeled in forms of RBFN and RNNs [7, 8]. However, these methods have inherent limitations in capturing long-term information as they adopted RNNs that require predefined time step configurations. With LSTM, our proposed method overcomes the aforementioned problems by preventing the gradient vanishing problem, thereby making accurate predictions based on variable length of temporal tool wear conditions. Recently, similar researches with LSTM have also been conducted for RUL prediction in various different scenarios [9, 10], the LSTM-based RCNN [11], has especially been applied to make RUL prediction for a similar setting of milling cutter. The proposed method integrates probabilistic prediction and can handle uncertainties to a certain degree.

The objective of this study is to improve tool wear estimation and RUL prediction by utilizing an LSTM based RNN framework, which makes more effective use of model parameters to train system observation and transition models of tool wear processes. We utilize minimally intrusive vibration sensors to collect the signal online and help train the RNN model. We train and compare our LSTM models with other forms of RNN. The solution approach can be extended to assess other critical assets of a manufacturing machine. The organization of this work is as follows: first, we provide a brief overview of related work in estimating tool wear using neural networks. The gated RNN architecture and equations are laid out as the basis of our work. We detail our experimental setup and selected cutting conditions for tool wear diagnosis and prognosis.

We finally report on the ability of the method to provide real-time estimation, one-step and two-step look ahead predictions of tool wear state, as well as RUL prediction.

2 Related work

Neural networks (NN) were widely used for modeling and predicting tool wear because they can learn complex non-linear patterns without the need for an underlying data relationship. They have strong self-learning and self-adaptive capabilities. Most previous works do not consider the sequential nature of the sensor data or are reliant on a combination of intrusive sensors that limits its practical use. Feature extraction and selection are most commonly performed, which therefore require some form of human labor involved in deciphering the sensor signals.

Wang et al.'s [5] work proposed the Fully forward connected neural network (FFCNN), and the optimized FFCNN by pruning some weights among the network. It was trained by the extended Kalman filter (EKF), rather than the backpropagation (BP). The result shows that it has better performance than Multilayer Perceptron (MLP) trained with BP. Özel et al. [6] applied the feed-forward neural network (FFNN) to estimate the surface roughness, and tool flank wear using a variety of cutting conditions. Bayesian regularization with Levenberg–Marquardt method is used in training to obtain good generalization capability and help in determining the number of neurons in the hidden layers. However, these two methods still did not consider time dependencies of the inputs and outputs to model, and cannot make prediction of the future tool wear.

Kamarthi et al. [7] and Luetzig et al. [8] proposed a method composing radial basis function networks (RBF) networks and an RNN. Different from other NN architectures that assume each input to be independent, RNNs such as [12–14] are designed with special units to record internal states over temporal sequences, which enables

itself to make predictions based on both current and historical inputs. RBFN approach fuses and maps the sensor signal to the tool wear, which leads to the estimation of tool wear amounts. The method considers the time dependencies of the inputs, the sensor signal, and the past estimated tool wear, which is better than the FFNN and MLP methods. However, the number of time steps of temporal dependencies is predefined in the model, rather than determined by any input value. Moreover, the developed method is limited to estimating the tool wear amount online, but cannot predict the future tool wear and hence the RUL of the cutting tool.

Overcoming the limitation of simple RNN, the gated recurrent neural networks were proposed in tool wear monitoring and prediction. The simple RNN can only realize one-step ahead and two-step ahead prediction, but its horizon on prediction is limited, meaning that it is limited in its ability to be extended to RUL prediction. The conventional RNNs cannot capture long term-dependencies, resulting in the NN not able to effectively capture previous long-term history into the current estimation. Specifically, the effect of an early-layer input on the final output diminishes rapidly as the layer of the NN increases in gradient-based training methods. This is referred to as the well-described vanishing gradient problem and it emerges especially in RNN architectures due to the large number of layers unfolded in the time-space [2, 15]. Therefore, gated recurrent neural networks, including LSTM and GRU, were developed to avoid this problem. Particularly, in LSTM, multiple gates have been introduced in each node to control the information flow. The self-connected recurrent edge (shown in Fig. 1), usually referred to as a constant error carousel, has a fixed weight of one which allows the gradient to pass through over many time steps without vanishing. BRNN, another variation of RNN, was proposed in [16] with a bidirectional network architecture that allows the usage of both historical and future inputs to make predictions. Fortunately, it is also possible

Fig. 1 The overall workflow of our proposed method with three phases

to combine BRNN and LSTM [17], which has led to a new architecture called BiLSTM. Inclusion of future inputs in these methods have led to various successful applications in speech recognition and language modeling tasks. Nevertheless, for the specific problem concerned in this paper, LSTM as one direction RNN, is chosen for simplicity as temporal sequences in the future are not likely to affect previous tool wear conditions. Similar to other NNs that benefit from depth in space, it is also possible to apply multiple RNNs, or LSTMs in particular, at the same time. It has been shown in [18] that stacked LSTMs yield significantly better performance in speech recognition while multi-dimensional LSTMs [19] can deal with data that have more than one spatio-temporal dimension. (The core of gradient vanishing problem is really the partial derivatives of the activation function stays at small values most of the time (e.g., < 1), when time steps are large, those small values got multiplied to an exponential close to zero. Both LSTM and RELU are methods to mitigate the gradient vanishing problem, not a complete solution, as the problem can result from multiple different computations in the back-propagation process. RELU or LSTM only resolves part of the most critical issue based on the application. When time steps are large, it is more likely that the propagation path diverge into a vanishing situation.)

The motivation for this work is that a model-free approach that does not need an underlying analytical model and it would be a better approach to capture long-term dependencies, prediction in an arbitrary horizon and remaining useful life prediction for a non-linear time-variant process. The advantages of our approach over existing ANN and RNN approaches are:

- LSTM based RNNs capture time dependencies and make predictions over conventional NNs that are based on Feed Forward Neural Networks.
- The proposed approach does not have the limitation of limited prediction horizon period, which is the major drawback of time-delayed neural networks.
- Time delayed based RNNs, although can achieve time dependencies, the method cannot realize arbitrary time step-ahead and RUL prediction.
- Our approach of including a model System Transition and System Observation function separately can naturally realize diagnosis and prognosis within the LSTM based RNN architecture.

3 RNN space model construction

The proposed approach is a model-free method based on Recurrent Neural Networks by formulating a temporal-dependent relationship between in-process indirect

measurement and in-process real machine state, and the relationships between the real machine state themselves in the temporal domain. Due to its model-free nature, RNN can model the non-linear relationship between their inputs and outputs. Therefore, the assumption pertaining to the Markov model and linearity can be avoided. Long-term dependencies cannot be modeled by conventional RNN approaches. Here we adapt the Long Short-Term Memory (LSTM) and Grated Recurrent Units (GRU) to model long term dependencies for multi-step prognosis of tool wear processes. The overall workflow is of our proposed method is shown in Fig. 1, which can be divided into three major phases: data collection, model training and online estimation. Firstly, we iterate the workpiece cutting process to capture measurements from the sensor and microscope. In the next training phase, we build a system observation function and a system transition function with simple RNN, LSTM and GRU. Furthermore, the collected experimental data split into a training, validation and test set. We train the system observation function and the system transition function separately to help tune its individual parameters and to implement independent training protocols. More importantly, the independent training also allows us these functions to go through varying training iterations. A comparison will be made between RNN, LSTM and GRU for the system observation and system transition function respectively, and then the best RNN model will be selected for one of these functions. Finally, with the trained models, an online machine state estimation is realized by the indirect sensor measurement. Building on the idea of generative models [20], an arbitrary time step ahead prediction can be updated in real-time, as well as to predict Remaining Useful Life (RUL).

3.1 System transition function

For dynamic mechanical systems, the system state may have temporal dependencies. More specifically, the current system state is affected by its past system states. Therefore, the System Transition function can be generically written as follows:

$$y_{t+1} \sim p(y_{t+1}|y^t; \theta), \quad \text{where } y^t = (y_1, y_2, \dots, y_t)$$

here y_t is the direct measurement of the system state. Without loss of generality, we study the operation of a milling machine system, where the cutter tool degrades over time when new cuts are conducted. We model each time step t as the sequence of cuts made until the tool wears out. In this case, we define y_t as the actual degradation state of the system, or the tool wear length at time t which can be measured with tools such as a microscope. Obviously, the system state changes as each cut is conducted

with dependency on conditions after previous cuts. When the system transition function is modeled as an RNN, the function can be rewritten as:

$$y_{t+1} \sim p(y_{t+1}|y_t, h_t^{trans}; \theta)$$

where h_t^{trans} is the hidden nodes for the system transition function, with superscript letter 'trans' denoting it. The current hidden node h_t^{trans} is a function of the past hidden node h_{t-1}^{trans} and current input y_t . Therefore, via the hidden node, the current hidden node bear all the current and past information to the inputs.

$$h_t^{trans} = f(h_{t-1}^{trans}, y_t; \theta)$$

$$\text{so that } h_t^{trans} = g_t^{trans}(y_1, y_2, \dots, y_t; \theta),$$

which the function g_t^{trans} takes the whole pass sequence $(y_1, y_2, \dots, y_t)$, so that the system state at time $t + 1$ can be modeled given all the past system state information. Note that in our implementation, the variable here is a continuous value and its distribution is assumed as unimodal, although it is possible to extend to the arbitrary value with arbitrary distribution.

3.2 System observation function

The direct measurement of the real-system state is often not practically feasible to be collected in production environments, such as actual tool wear amounts on cutting tool inserts. The system state can be inferred from indirect measurements, such as one or more sensor signals. Also, in most cases, there is not a good function that can build a relationship between the direct measurement of the system state and its associated indirect measurement. Also, any developed relationship must consider temporal effects. Specifically, with availability of any past direct measurements of the system state, the determination of the current system state should refer to all past indirect measurements. Therefore, the system observation function can be written as:

$$y_t \sim p(y_t|x^t, \theta), \quad \text{where } x^t = (x_1, x_2, \dots, x_t)$$

here x_t is the indirect measurement at time step t ; and y_t is the direct measurement of the system state.

With RNN, the system observation function can be rewritten as:

$$y_t \sim p(y_t|x_t, h_t^{obs}, \theta)$$

where h_t^{obs} is the hidden nodes for the observation function with superscript letter 'obs' denoting it.

Similar to h_t^{trans} , the hidden nodes for the system transition function,

$$h_t^{obs} = f(h_{t-1}^{obs}, x_t; \theta)$$

so that $h_t^{obs} = g_t^{obs}(x_1, x_2, \dots, x_t; \theta)$, which the function g_t^{obs} takes the whole pass sequence $(x_1, x_2, \dots, x_t)$.

3.3 Long short term memory (LSTM) cell

LSTM cell is an advanced version of the RNN cell, which is excellent at remembering values for either long or short time periods. It consists of repeating memory cell units, each of which consists of interacting elements—the cell state, and three gates that protect the cell state, input gate, a forget gate and an output gate (Fig. 2).

The gates, via the activation function, control the flow of information to update the cell state. The input gate controls the new input flow into the memory, and the output gate controls the extent of the flow from memory into the

Fig. 2 LSTM cell block diagram. The forget gate acts like “throttle” controlling the amplitude of the input, output, and self-loop, respectively

output nodes. The forget gates, which implements in the self-loops, control the gradient flow from previous cells from the long temporal past, thereby adaptively forgetting or including the distant memory. Additionally, with gated self-loops controlled by the weights, the time scale of integration can also be controlled by the forget gates [21]. At each time step t , the hidden state, h_t is updated by x_t , the current data at the time step, the hidden state at previous time h_{t-1} , input gate i , forget gate f , output gate o and the internal memory cell c . U and W are the weight matrices. input gates:

$$i_t = \sigma_i(x_t U^i + h_{t-1} W^i)$$

forget gates:

$$f_t = \sigma_f(x_t U^f + h_{t-1} W^f)$$

output gates:

$$o_t = \sigma_o(x_t U^o + h_{t-1} W^o)$$

$$g_t = \tanh(x_t U^g + h_{t-1} W^g)$$

where g is candidate hidden state which is used to calculate the real hidden state.

$$c_t = c_{t-1} f_t + g_t \cdot i_t$$

$$h_t = \tanh(c_t) o_t$$

As seen from the above equations, all the gates can be determined by the input x_t and hidden nodes h_{t-1} , with weights that can be tuned during the training process.

Finally, the output calculation, the output y_t is written as:

$$y_t = \sigma_y(h_t W^y + b_y)$$

where σ is activation function, $\cdot$ is the Hadamard product, x_t is the input at time step t ; and y_t is the output at time step t , and b is bias.

3.4 Gated recurrent units (GRU) cell

GRU cell is a special kind of gating mechanism for the LSTM cell of an RNN. It has a simpler structure and thus the fewer parameters than the LSTM cell. As a simplified version of LSTM cell, the GRU cell was proposed by [22], which has less gates and thus fewer weights to be tuned. There are only two kinds of gates for GRU: reset gate, r , and update gate, z . The reset gate determines how to combine new input with that of the previous memory in its cells. On the other hand, the update gate decides the portion of its

previous memory does it retain or throw out. The equations for the GRU cell are similar to the LSTM

$$z_t = \sigma_z(x_t U^z + h_{t-1} W^z)$$

$$r_t = \sigma_r(x_t U^r + h_{t-1} W^r)$$

$$n_t = \tanh(x_t U^h + r_t \cdot h_{t-1} W^h)$$

$$h_t = (1 - z_t) \cdot n_t + z_t \cdot h_{t-1}$$

Since there are only 2 gates, the important difference between a GRU versus an LSTM is that GRU does not possess an internal memory (c_t). The reset gate is directly applied to the previous hidden state (h_{t-1}). Thus, it can train faster since there are only few parameters in the U and W matrices and hence may be useful when not much training data is available.

3.5 RNN training process with long term dependencies

The *Early Stopping* technique is used for the training process. the number of iterations is the count of looping through the whole training set, and the Mean Square Error (MSE) of the training set and validation set is estimated for each loop. The MSE of the validation data set is used as a reference to meet certain criteria about the MSE of the system state estimation. For each time step, the weights in the RNN model are updated by the *Back Propagation Through Time (BPTT)*, which is a gradient-based technique to train the RNN. Specifically, the Adaptive Moment Estimation (Adam), having the best performance is used for the BPTT [23]. Since the BPTT multiplies gradient over a large number of time steps, the resulted gradient either get vanished or explode over multiple layers. The recurrent net tends to induce LSTM cell and GRU cell to avoid the gradient vanishing occurrence in simple RNN. On the other hand, the exploded gradient can lead to the training process to be unstable, leading to the algorithm to find a non-optimal solution. This is solved by *clipping the gradient* to a determined maximum value. As stated previously, the RNN for System Transition and System Observation are trained independently. The RNN cells (Elman RNN cell), LSTM cell, and GRU cell will be tested and compared on the performance index of their online estimation. The Mean Square Error (MSE) will be applied as a criterion for these three different cells. Moreover, as model-free methods, there is no assumption that there are any certain forms of relationship between the variables at the time domain, regardless of whether they are linear or nonlinear.

3.6 Diagnosis and prognosis with RNN

After the three kinds of RNN model of System Transition and System Observation is trained, they can be further applied for diagnosis and prognosis of the system state, by inputting the indirect measurement in real-time. Our proposed method is to realize online diagnosis, by taking advantage of the indirect measurement from past and current states: $p(y_t, \theta | x^t)$, where $x^t = (x_1, \dots, x_t)$. For prognosis, following the method of the generative recurrent neural networks [20], the produced values from output nodes and the hidden nodes of the RNN of the System transition are inputted into the input node and hidden node of the model itself. This process is repeated, and the number of the repetitions is equal to the predicted number of time step horizons. Lastly, for Remaining Useful Life (RUL) prediction, the criterion of the degradation, can be predefined to a certain value, for example 0.3 mm for the tool wear. By applying the generative model of RNN for system transition (which means if we input the values of the hidden and output nodes of RNN for system transition of the last time step into the hidden nodes and input nodes of the next time step, and repeat the process), and consider the error from the observation model and the error propagation over the transition model (it subjects to the Wiener process [24] that error accumulate over time step), until the certain percentile confidence [25] (95 percentile is used in this paper) of predicted degradation variable reaches the criterion, the number of time steps it passes through are used as the predicted RUL, which in our case, will be the number of time steps it will take to reach the tool wear criteria.

4 Experimental evaluation

4.1 Setup

To test and verify the LSTM based RNN method in tool wear diagnosis and prognosis, a similar experiment setup as the previous work [26] is applied. The milling machine, HAAS VF2 CNC is used. The cutting process is the dry milling without coolant liquid. The Steel 4142 (cold rolled, 40-45HRC) is used as the workpiece material. Sandvik Coromill 390 Insert (390R-070204 M-PM) as one insert was installed into a two-flute indexable tool (12.7 mm diameter) Sandvik Coromill 390 (RA390-013013-07L) and the tool is installed in a tool holder. The milling cutter repeatedly cut the workpiece with the same conditions: cutting length for a cut is 37.0 mm, spindle speed is 500 rounds per minute. The axial depth of the cutting is 0.4 mm and the radial depth of cutting is 6.5 mm for the milling cutter. The feed rate between the cutter and workpiece is 50 mm/min. These condition results in the surface speed for the cutting is 19.81 m/min and chip load keep at 0.05 mm. The accelerometer, Kistler accelerometers 8762A10, was installed onto the fixture for the workpiece (Fig. 3) to pick up the vibration RMS signal during the cutting process. The accelerometer is used because it is not an intrusive sensor like the force dynamometers. During the repeated cutting process, after each cut, the microscope was used to measure the tool wear. When the tool wear reaches or surpasses the predefined threshold, 0.3 mm, the new cutter will replace the current cutter. There are two samples of the process of the RMS value of acceleration signal in Fig. 4. The values do not monotonically increase so that there are could be dependency of the previous cutting for one cut.

Fig. 3 Schematic diagram of the experimental setup with micrographs of the cutting tool surface at various cut path numbers

Fig. 4 Vibration Index for two replications of cutting the same material under similar machining conditions. The only difference is that each replication was carried out by a brand new cutting tool insert but of the same specifications

4.2 Data collection

The data that we have collected for validation of our proposed method primarily source from the vibration sensor (Kistler accelerometer) and microscope. The sensor, as described above, outputs RMS (Root Mean Square) signal information, which is a common metric to determine machine vibration. Meantime, after each cut, we measure the tool wear length (mm) with microscope. Figure 3 shows an illustration of tool wear degradation after multiple cuts were conducted. We repeat the data collection process 15 replications, and then feed 13 replications of the RMS vibration data and measured tool wear length information into our models for training and validation. After that, the trained model can be deployed online to predict tool wear degradation and RUL based on real-time vibration sensor signals. Note that the sequence lengths of different training datasets may vary due to the uncertainty of the tool wear process, each of the sequences ends when tool wear reached the 0.3 mm threshold.

4.3 Model training

Early stopping is applied to loop through the training set for a number of iterations for the three kinds of RNN model for the system transition and observation functions, respectively. The mean square error (MSE) of the validation set is evaluated for each iteration. The whole dataset is divided into three parts, the first one is the training set (7 pairwise sequences), the second one is the validation set (6 pairwise sequences), and the last one is the test set (2 pairwise sequences). For each time step, the weights in the RNN model (U and W) are updated by the backward propagation through time (BPTT). For each sequence,

the values of hidden nodes are reset to zero. The RNN for System Transition and System Observation is also trained independently.

During the training process, the reduction rate of the average MSE decreases as the training proceeds. The iteration number is set as 14, given the computational capacity and average MSE are good enough for estimation. Comparing the MSE values for the System Transition function, the results show that the average MSE of the LSTM is lower than the other two—Elman RNN and GRU. The Standard deviation (Std.) also shows that the stability of the LSTM to achieve relatively good performance for both system transition and observation (see Table 1).

All the RNNs have one hidden layer with 5 hidden nodes. The MSE of validation set of trained LSTM for System Transition function is 0.000736 mm², and the MSE of validation set of applied trained LSTM of the System Observation function is 0.002529 mm². Since there are more parameters in the LSTM cell than the other two, Elman RNN and GRU, it shows its flexibility to fit the data and exhibit more adaptive properties to construct the complexity of the time dependencies, although more computational resources may be needed for determining the parameters in the model.

5 Results

5.1 Online diagnosis

By applying the LSTM for system observation, given the indirect measurement of the current and past time step, the RMS of the vibration signal, the model can estimate the tool wear at the current time step.

Table 1 Compare MSE of validation set for RNN, LSTM, and GRU for System Transition and Observation

RNN cell type	System transition		System observation	
	Avg. of MSE (mm ²)	Std. of MSE (mm ²)	Avg. of MSE (mm ²)	Std. of MSE (mm ²)
Elman RNN	0.000977	0.000978	0.002509	0.001385
LSTM	0.000736	0.000112	0.002529	0.001155
GRU	0.000946	0.000686	0.001979	0.001501

Fig. 5 Online Tool Wear Diagnosis with indirect measurement. The dots indicate the actual measured tool wear; the solid lines are the estimated tool wear. **a** Replication 1; **b** Replication 2

Table 2 Avg. metric value for the test set

Cell type	MSE y	MAPE y (%)	MSE RUL	MAPE RUL (%)
Elman RNN	0.000655	9.70	16.3	49
LSTM	0.000478	8.35	5.4	30
GRU	0.000485	7.90	19.5	53

For Replication 1 (Fig. 5A), the estimated tool wear closely fit the real tool wear, and thus the curve shape of the estimation appears more like a bending line, rather than a smooth curve, thanks to the online estimation of the tool wear which is adjusted based on the indirect measurement. For the Replication 2 (Fig. 5B), there is an abrupt tool wear change between the 3rd to the 4th cut. The estimated tool wear adjusts for approaching the real tool wear, with certain time lag. This illustrates that the method is robust to certain extent on tolerating abrupt change. The result in Table 2 also shows that the LSTM for the tool wear y has the best value among the tree cell types in terms of MSE and MAPE.

5.2 Online one-step and two-step ahead prediction

For one-step ahead prediction, the estimated tool wear at the current time step is input into the input node of the LSTM for the System transition function. It assumes

that the tool wear at time step 0 is equal to 0.0 mm ($y_{t=0} = 0.0$ mm). The predicted tool wear at next time step $t + 1$ is the value from the output node of the RNN for the System transition, given the hidden node and estimated tool wear at the current time step t . This means that the tool wear at the next time step is predicted given the current and past tool wear estimation as given by the system observation function (Fig. 6). Similar to online one-step ahead prediction, the tool wear at two time-step ahead is predicted given the next, current and past tool wear estimates by the system transition function and the system observation function and (Fig. 7). The predicted tool wear for one-step and two-step ahead appears smoother than online estimated ones, and does not closely fit the tool wear. They provide the prediction of the tool wear and show the trend of the tool wear. When the time step proceeds to the predicted cutting step, the estimation is updated according to the online indirect measurement.

The proposed method is robust to certain abrupt changes in the system state, the tool wear. An arbitrary number of time step ahead prediction can be realized by the proposed method. Similar to online two-step ahead prediction, the tool wear at three or multiple time steps ahead can be predicted given the two-step ahead or multiple time minus one step ahead, current and past tool wear estimation by the system transition function and system observation function respectively.

Fig. 6 Online Tool Wear One-Step Ahead Prediction with indirect measurement. The solid circles indicate the actual tool wear; the solid lines are the predicted tool wear

Fig. 7 Online RUL Prediction. The dashed lines indicate the actual RUL; the solid lines are the predicted RUL

5.3 Remaining useful life (RUL) prediction

At each time step, by looping the transition model, given the error from the observation model and error accumulation, according to the Wiener process, of the transition model over the number of loops, until the 95 percentiles of value of the output nodes value reaches the criterion for the tool wear set to be at 0.3 mm. Both of the RUL predictions start at 14 cuts (Fig. 7). For replication 1 and 2, the predicted RUL is updated according to the online indirect measurement, so that the predicted RUL can be adjusted to approach the real RUL. The result in Table 2 also shows that the LSTM for the RUL has the best value among the three cell types in terms of MSE and MAPE.

6 Discussion

LSTM and GRU, as gated RNN, can capture the long-term time dependencies as a model for the system transition and observation functions, compared to the simple RNN and Elman RNN. By taking advantage of the trained RNN system observation model, the tool wear can be estimated from currently observed indirect measurement and past indirect measurements. Also, for tool wear prediction, one-step ahead tool wear prediction can be realized by inputting into the RNN system transition function, the estimated current and past tool wear by the RNN system observation function. For two-step ahead tool wear prediction, the one step ahead tool wear prediction by the RNN system transition function and all the estimated tool wear by the RNN system observation function act as inputs into the RNN system transition model, thus, resulting in the two step ahead tool wear prediction. Similarly, the multiple time step ahead prediction can also be estimated. This is attributed to the design of the generative RNN, in which our proposed model can predict an arbitrary number of time step ahead tool wear degradation in the future.

Two replications of the tool were tested to demonstrate and evaluate the proposed model approach. It is to be noted that despite the same cutting conditions,

the tool wear diagnosis and prognosis are quite different for the two replications. This is typical of all cutting tools since they can wear down at varying rates and patterns. We are able to predict the tool wear amounts on the fly, which confirms that the necessary diagnosis and prognosis methods can be adapted online. Furthermore, the RUL prediction can also be predicted until the tool wear reaches the set criterion.

In this study, the indirect measurement is the signal obtained from the vibration generated during the cutting process. As a black box method, the Neural Network model does not explicitly show the formula of the system observation and system transition function as seen in the Particle Learning method proposed by Zhang et al. [26] and any other analytic models typically used in cutting tool wear estimation. The LSTM based RNN approach can increase the computation cost, and the challenge here is to decide the complexity of the network, the number of layers and number of hidden nodes, given the reward is the flexibility of the fitting any form of relationship. Additionally, the system transition function also needs to be trained beforehand before applying the prognosis process. This method can have desirable diagnosis and prognosis results especially if the tool wear gradually occurs but also can tolerate certain abrupt tool wear change. However, for different materials and different machining conditions, new data should be collected to train a new model.

The LSTM based RNN can make predictions more superior than the feed-forward neural network. Second, the approach is also better than the time-delay neural network, which can only predict limited time step ahead of machine states. Third, for the RNN with only time-delayed form, it cannot realize the arbitrary number of time step ahead prediction, not to mention not being able to predict RUL. Fourth, Malhotra et al. [27] used the RUL as target variable and sensor signal to be modeled with the RNN. However, the RUL does not reflect any physical characterization, so that there is no direct physical connection between the sensor signal and RUL. Also, the approach of using RUL as the target variable cannot realize machine state diagnosis and prognosis. Compared to these

methods, the approach of modeling the system transition function and system observation function with the RNN renders the online estimation considering the time dependencies become natural. Taking the schema of the generative RNN, arbitrary time step ahead prediction gets realized by combining the system transition function and system observation function modeled by RNNs. Further, by applying advanced form RNN cell, the LSTM, GRU, the long-term time dependencies can be captured.

The neural network model is a more flexible method than the analytic model, which does not need physical formula within the model. The proposed approach by modeling the system transition and system observation with long-term time dependency helps realize the online estimation of the system state and predicts the arbitrary future horizon of the system state, by the given current and past indirect measurements. The application of this method can be extended into any critical components of the physical machine, such as bearings, gears, engine, ball screws, grinding wheels, nano-machining processes and bioreactors, in which only indirect measurements are accessible in real-time and there is a need to estimate and predict the system state. Additionally, the proposed model can be extended and applied for multi-fault components. In many cases, a single multi-layer NN would be sufficient to predict faults including different causes while it is also possible to train separate NNs for more interpretable results in complex problems. In certain applications, the input data volume can be considerably huge, for example, when the input are high dimensional or involve very long temporal sequences to learn. For effective learning in big data scenarios, it is common to use dimension reduction methods such as Principal Component Analysis (PCA) [28, 29], or seek parallelization to speed up the training process with optimized GPUs like NVIDIA CUDA [30, 31]. Interestingly, when little data are present for a certain problem, it is even viable to transfer the knowledge from another domain to build a NN model in the target domain [32].

There are several possible ways to improve the accuracy of the prediction. The first one is to loop through more number of iterations of the training set to train the model; the second one is to use more hidden nodes or more layers of hidden nodes and even combining the dropout technique [33] within the neural network model; the third one is to collect more data to train the model. With more data to train the model, and more available data for the validation, the model can get more robust and have more stable performance in diagnosis and prognosis. Moreover, one can also use the n-fold or leave-one-out cross-validation method to get a stable evaluation of the MSE of both the RNN for system transition and RNN for system observation. Incorporating more time domain and frequency domain feature of the signal into the inputs of the model,

so that feature engineering work should be implemented. Uncertainty is another important metric to consider for any given prediction model. Without proper handling, it would be difficult to establish confidence in prediction results and make meaningful prognosis recommendations. Therefore, we conducted repeated experiments with standard metrics (MSE, MAPE) which empirically prove our model to make accurate predictions with low deviations. This can be further improved by incorporating the evaluation framework proposed in [25], which articulates a standardized process to handle uncertainties in model predictions with comprehensive representations such as probability distributions and fuzzy membership functions. One viable solution is to obtain probabilistic information in RUL prediction with variational inference proposed in the RCNN-based method [11], however, we leave that as a future work to incorporate in any following prediction models due to space limitations.

7 Conclusion

The paper proposes a framework for the machine state diagnosis and prognosis using RNN models for system transition and observation function via indirect measurements. The LSTM and GRU, which can capture the long-term dependencies are used to model those two functions. The validation set was used to evaluate the estimation accuracy by the MSE. Two replications of the indirect measurement sequences, as the test set, are used by the proposed model, to demonstrate the online estimation, one-step prediction, two-step prediction, and RUL prediction. The model results, including both tool wear estimation and the RUL prediction, were highly accurate as compared with the real-world measurements. The validation set shows acceptable MSE for both system observation and transition functions. After comparison with the real values, the results show good performance and a strong possibility for practical usage. First, as model-free methods, the Neural Network requires no analytic formula in modeling. Second, the LSTM cell modeled system transition and system observation functions, which capture the long-term temporal dependencies, and achieved better performance than the simple RNN to model them. The arbitrary horizon of prediction can be realized by taking advantage of the modeled system transition function and the generative model, and thus the RUL prediction. Despite the high prediction accuracy of our proposed method, several aspects that remains to be investigated as future work:

Model robustness and generalizability our current experiments focused on a specific setting where relatively short-term sequence learning with only a few cuts

is involved. More improvements may be done towards a fully-generalizable and robust LSTM model if the experiments can be extended to include more data like event information and other scenarios such as dealing with much longer temporal sequences for continuously-running industrial tools. As shown in our experiments, our current model is robust to some extent, as it can learn from certain abrupt changes. Nevertheless, real-world signals may contain false alarms due to sporadic sensor failures, which diminishes the quality of training data. This is in fact an open NN question known as *garbage in garbage out*. In order to train NN without overfitting these false alarms, or noisy labels, noise-robust loss functions [34] and sample reweighting [35] can be applied. Dynamic loss correction in the training process is also possible if the data corruption probability is known a priori [36]. Moreover, removal and imputation of outliers that can be identified easily may be a simple but cost-effective approach in many cases.

Automated close-loop diagnosis and prognosis

the current workflow required for data collection, model training and inferencing is quite complicated for further applications into real-world environments. With a built-in microscope camera for image data collection, a fully-automated system can be built where the tool wear condition can be captured and inferred possibly by a CNN to feed into our LSTM model for real-time diagnosis and prognosis with RUL estimation into the future.

It is very possible to extend the proposed methods for diagnosis and prognosis of system state by indirect measurement for other physical entities, such as gears, bearings, ball screws, motors, etc.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Jardine AK, Lin D, Banjevic D (2006) A review on machinery diagnostics and prognostics implementing condition-based maintenance
- Hochreiter S, Bengio Y, Frasconi P, Schmidhuber J (2001) Gradient flow in recurrent nets: the difficulty of learning long-term dependencies
- Werbos PJ (1990) Backpropagation through time: what it does and how to do it. *Proc IEEE* 78(10):1550–1560
- Hochreiter S, Schmidhuber J (1997) Long short-term memory. *Neural Comput* 9(8):1735–1780
- Wang X, Wang W, Huang Y, Nguyen N, Krishnakumar K (2008) Design of neural network-based estimator for tool wear modeling in hard turning. *J Intell Manuf* 19(4):383–396
- Özel T, Karpaz Y (2005) Predictive modeling of surface roughness and tool wear in hard turning using regression and neural networks. *Int J Mach Tools Manuf* 45(4):467–479
- Kamarthi SV, Kumara SRT (1995) A new neural network architecture for continuous estimation of flank wear in turning. In: *Proceedings of the 1st world congress intelligent manufacturing processes and systems* (vol 2, pp 1145–1156)
- Luetzig G, Sanchez-Castillo M, Langari R (1997) On tool wear estimation through neural networks. In: *International conference on neural networks* (vol 4) IEEE, pp 2359–2363
- Khazaee M, Banakar A, Ghobadian B, Mirsalim MA, Minaei S (2020) Remaining useful life (RUL) prediction of internal combustion engine timing belt based on vibration signals and artificial neural network. *Neural Comput Appl* 1–17
- Yu Y, Hu C, Si X, Zheng J, Zhang J (2020) Averaged Bi-LSTM networks for RUL prognostics with non-life-cycle labeled dataset. *Neurocomputing*
- Wang B, Lei Y, Yan T, Li N, Guo L (2020) Recurrent convolutional neural network: A new framework for remaining useful life prediction of machinery. *Neurocomputing* 379:117–129
- Jordan, M.I., 1997. Serial order: A parallel distributed processing approach. In: *Advances in psychology*, vol 121. North-Holland, pp 471–495
- Hopfield JJ (1982) Neural networks and physical systems with emergent collective computational abilities. *Proc Natl Acad Sci* 79(8):2554–2558
- Elman JL (1990) Finding structure in time. *Cogn Sci* 14(2):179–211
- Bengio Y, Simard P, Frasconi P (1994) Learning long-term dependencies with gradient descent is difficult. *IEEE Trans Neural Networks* 5(2):157–166
- Schuster M, Paliwal KK (1997) Bidirectional recurrent neural networks. *IEEE Trans Signal Process* 45(11):2673–2681
- Graves A, Schmidhuber J (2005) Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Netw* 18(5–6):602–610
- Graves A, Mohamed AR, Hinton G (2013) Speech recognition with deep recurrent neural networks. In: *IEEE international conference on acoustics, speech and signal processing*. IEEE, pp 6645–6649
- Graves A, Fernández S, Schmidhuber J (2007) Multi-dimensional recurrent neural networks. *International conference on artificial neural networks*, Springer, Berlin, Heidelberg
- Graves A (2013) Generating sequences with recurrent neural networks. arXiv preprint <http://arxiv.org/abs/1308.0850>.
- Goodfellow I, Bengio Y, Courville A (2016) *Deep learning*. MIT Press
- Chung J, Gulcehre C, Cho K, Bengio Y (2014) Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint <http://arxiv.org/abs/1412.3555>.

23. Kingma D, Ba J (2014) Adam: a method for stochastic optimization. arXiv preprint <http://arxiv.org/abs/1412.6980>.
24. Si XS, Wang W, Hu CH, Zhou DH (2014) Estimating remaining useful life with three-source variability in degradation modeling. *IEEE Trans Reliab* 63(1):167–190
25. Saxena A, Celaya J, Saha B, Saha S, Goebel K (2010) Metrics for offline evaluation of prognostic performance. *Int J Prognost Health Manag* 1(1):4–23
26. Zhang J, Starly B, Cai Y, Cohen PH, Lee YS (2017) Particle learning in online tool wear diagnosis and prognosis. *J Manuf Process*
27. Malhotra P, Tv V, Ramakrishnan A, Anand G, Vig L, Agarwal P, Shroff G (2016) Multi-sensor prognostics using an unsupervised health index based on lstm encoder-decoder. arXiv preprint <http://arxiv.org/abs/1608.06154>.
28. Pearson K (1901) LIII: On lines and planes of closest fit to systems of points in space. *Lond Edinb Dub Philos Mag J Sci* 2(11):559–572
29. Hotelling H (1933) Analysis of a complex of statistical variables into principal components. *J Educ Psychol* 24(6):417
30. Raina R, Madhavan A, Ng AY (2009) Large-scale deep unsupervised learning using graphics processors. In: *Proceedings of the 26th annual international conference on machine learning*, pp 873–880
31. Harris M (2008) Many-core GPU computing with NVIDIA CUDA. In: *Proceedings of the 22nd annual international conference on Supercomputing*, p 1
32. Pan SJ, Yang Q (2009) A survey on transfer learning. *IEEE Trans Knowl Data Eng* 22(10):1345–1359
33. Srivastava N, Hinton GE, Krizhevsky A, Sutskever I, Salakhutdinov R (2014) Dropout: a simple way to prevent neural networks from overfitting. *Journal of machine learning research* 15(1):1929–1958
34. Ren M, Zeng W, Yang B, Urtasun R (2018) Learning to reweight examples for robust deep learning. arXiv preprint <http://arxiv.org/abs/1803.09050>.
35. Zhang Z, Sabuncu M (2018) Generalized cross entropy loss for training deep neural networks with noisy labels. In: *Advances in neural information processing systems*, pp 8778–8788
36. Patrini G, Rozza A, Krishna Menon A, Nock R, Qu L (2017) Making deep neural networks robust to label noise: a loss correction approach. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp 1944–1952

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.