AEDmts: An Attention-Based Encoder-Decoder Framework for Multi-Sensory Time Series Analytic

JIN FAN, HONGKUN WANG, YIPAN HUANG, KE ZHANG, AND BEI ZHAO

Department of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou 310018, China
Corresponding author: Bei Zhao (zhaobei@hdu.edu.cn)

This work was supported in part by the Key Science and Technology project of ZheJiang Province under Grant 2019C03127.

ABSTRACT Numerous IoT applications have emerged in human healthcare area with advances in wearable electronics. Multiple physical and physiological data bearing strong spatial-temporal characteristic collected by the wearable sensors is sent to the smartphones where it is aggregated and transferred to back-end applications for further processing. Analyzing the multivariate time series is of great importance yet very challenging as it is affected by many complex factors, i.e., dynamic spatial-temporal correlations and external factors. In this paper, we propose an attention-based encoder-decoder framework for multi-sensory time-series analytic. It consists of four parts: data collection, data mining, time-series analytic part and user interaction. A temporal-attention based encoder-decoder model is proposed to make a long-term prediction of multiple time series to realize the real-time user interaction. The proposed model uses the LSTM model to learn the long-term dependence of the time series related to certain motion sequence. The attention mechanism connects the encoder and the decoder to make long-term predictions for future time series. Through extensive experiments, the proposed model has achieved better results in short-term and long-term predictions compared with the state of art methods. An activity recognition algorithm based on LSTM is also proposed in this framework to identify daily human activities and sports activities accurately. Through five-fold and ten-fold cross-validation strategies and comparison with six baseline machine learning models, the activity recognition algorithm has a recognition rate of 98.89% and 99.28% for human activity.

INDEX TERMS Encoder-decoder framework, multi-sensory time series analytic, motion prediction.

I. INTRODUCTION

The recent developments in machine learning offers great opportunities to realize intelligent applications over human healthcare. With the repaid development of Internet of Things, massive physical and physiological data bearing strong spatial-temporal characteristic is generated. It is of great importance to mine these time series data to guide sport exercises or monitor the daily activities. Researches related to time series data have played an important role in many fields, such as traffic flow prediction [1], stock prediction [2], activity detection and classification [3], [4], human-computer interaction [5], social network real-time recommendation [6], [7] etc. The motion data collected by the wearable device’s acceleration and angular velocity sensors is presented in a form of multivariate time series with a certain pattern. Generally, the time series prediction method is based on the historical observations $y_1, y_2, y_3, ..., y_{T-1}$ to generate future sequences $y_T$, but ignores the impact of exogenous sequences $x_1, x_2, x_3, ..., x_{T-1}$ on the final prediction results. These methods usually predict a future value at certain time steps. For multivariate sequences long-term prediction, it is becoming a challenging problem, as it is extremely difficult to extract related exogenous sequences dependencies for long-term prediction. Recently, a recurrent neural network model based on the double attention mechanism [8] can successfully extract the spatio-temporal dependence of multivariate exogenous sequences to predict $y_T$, but this method only performs well in the next step of prediction. Traditional RNNs suffers from the vanishing gradient problems so that they can not capture the long-term dependencies for time series data. The encoder-decoder network originated from sequence to sequence model becomes popular as its successful applications in natural language processing. However,
the performance of encoder-decoder networks will deteriorate rapidly as the length of input sequence increases, which presents major challenge to long-term motion prediction. In order to solve this issue, we develop a temporal attention based encoder to decoder model for multivariate time series long-term prediction (Multi-TAED), which can realize real-time motion prediction.

On the other hand, many researchers have designed wearable sensor device-based recognition systems through data pre-processing and machine learning classification algorithms [9]–[15]. These methods can achieve acceptable classification results with the supervised training process. However, these methods are mainly offline trained and the accuracy can be further improved. With the rapid development of wearable devices, deep learning methods can be implemented on resource-constrained devices (such as a smartphone) to give real-time feedback, which could reveal the motion characteristics that are closely related to human dynamics, from simple action to complex motion recognition.

To this end, this paper proposes an attention-based encoder-decoder framework for multi-sensory time-series analytic of wearable sensor devices. It integrates a multivariate time series long-term prediction algorithm (Multi-TAED) and an activity recognition algorithm based on deep learning method. The wearable sensors are placed on the user’s two arms and on the upper sides of both knees and can perform daily activity measurements and collect motion signals without relying on any external equipment and venue requirements. The proposed Multi-TAED algorithm can make long-term predictions of time-series signals, and compare the predicted time series with ground-truth to give users real-time feedback. In addition, the framework also provides an activity recognition algorithm to identify daily human motion.

Our contributions of this work lies in:

1) In this work, a wearable sensor system is proposed, in which a time-series prediction algorithm (Multi-TAED) and an activity recognition algorithm are integrated. This prototype could collect multi-sensor data through human motion and offer real-time interactions to users.

2) A temporal-attention based encoder-decoder model for multivariate time series long-term prediction (Multi-TAED) is designed for motion prediction. The performance of Multi-TAED outperforms the state-of-art methods in terms of the root mean square error under the different prediction time-step scale.

3) The proposed activity recognition algorithm utilizes the LSTM model as a classifier. Through extensive experiments, the performance in terms of accuracy is better verified than other machine learning model-based algorithms.

The rest of this article is organized as follows: Section II introduces the related work of motion prediction. The proposed wearable sensor system is presented in Section III. Section IV introduces the ready to work and the experimental results and discussion of the two algorithms. Finally, the conclusions are given in Section V.

II. RELATED WORK

Our work is mainly related with two aspects: time series prediction and deep learning based activity recognition algorithms.

A. TIME SERIES PREDICTION

Continuous motion signals collected by an IoT system are multivariate time series data. The prediction of motion is essential to forecast the long-term time series. Among the classic models, ARIMA [16] is an essential method for studying time series. It consists of a mixture of autoregressive models (AR) and moving average models (MA). The model usually requires that the time series data is stationary and can only capture linear relationships. A nonlinear autoregressive index (NARX) model [17] establishes a mapping system from past input and output and independent noise to the future nonlinear output. But such methods are predefined nonlinear forms, so they cannot capture the true underlying state of a time series. The traditional regression algorithm of machine learning SVR [18] doesn’t pay attention to the correlation between time series. Recurrent neural networks(RNN) [19] have been successfully used in sequence learning. But traditional RNNs will encounter gradients issues when learning long-term dependencies. Long short term memory unit (LSTM) [20] and gated recurring unit (GRU) [21] overcome this limitation. They are commonly used in time series prediction recently [22], [23]. Although LSTM and GRU could capture long-term dependencies, the model does not consider the impact of the dynamic characteristics of the exogenous sequence on the target sequence. The encoder-decoder model based on LSTM and GRU [24] has shown great success in language translation. The encoder-decoder model is also a universal end-to-end framework for sequential data processing. The encoder encodes the information into a fixed-length vector, and then decodes the vector into final prediction result. However, there is a problem with the encoder-decoder model. As the length of the input sequence increases, the overall performance of the model will decrease. Then, Cho et al. [25] proposed an attention-based encoder-decoder network. The attention mechanism requires the decoder to look back at the output of the encoder to find the information most relevant to the final prediction sequence.

B. ACTIVITY RECOGNITION ALGORITHM

Sensors such as accelerometers and gyroscopes are embedded in the wearable device, making it possible to collect motion signals of different body parts of the user. The received motion signals can be used to identify human activities. These activities include not only simple physical activities such as standing, walking, running, but also complex activities such as cooking and bathing. Researchers have effectively developed human daily activity recognition systems through pre-processing of motion signals combined with machine learning classifiers, and have achieved good research results. Chen et al. [9] proposed a classifier based on
the fuzzy basis function of an acceleration sensor to classify the daily activities of 8 people in the laboratory with satisfactory recognition accuracy.

With the rapid development of smartphones, most smartphones have built-in accelerometers, gyroscopes, digital compasses, and adequate computing abilities. A large number of researchers use smartphones as experimental platforms to activity recognition algorithm. Anjum and Ilyas [14] developed an application to identify seven types of sports for users based on machine learning models. Although smartphones sensors bring convenience to users, collected data is strongly affected by the position of phone placement. Nair et al. [15] use unprocessed raw data as input to the model and used a convolutional network-based model to classify different human activities. It improved the classification accuracy to a certain extent. In short, for complex data, the sensors built into smartphones cannot collect a large number of accurate data sets, so the performance of classification is poor.

III. SYSTEM

The framework of AEDmts is shown in Fig. 1. It includes three parts: data collection, server computing, and front-end visualization. The motion signals of the human body measured by the sensors are collected by the micro-controller of the wearable device. The slave device sends the signal to the host through the RF wireless transceiver. The host device then sends all the signals to the connected smartphone, where the designed App can give user real-time feedback based on the collected motion signal.

A. DATA COLLECTION

1) APPARATUS

The proposed wearable inertial sensing device is illustrated in Fig. 2. Wearable sensor devices are fixed to various parts of the tester’s body (two on the upper arm and two on the knee, with the left arm on the master and the rest on the slave) to collect motion signals from the device. The master sensor and three slave sensors include a microcontroller (STM32F103), a six-axis inertial sensor module (MPU-6050), a radio frequency wireless transmission module (nRF24L01), a Bluetooth module (HC05), and a power supply. There is no bluetooth module in slave. The microcontroller collects the motion signals measured from the six-axis inertial sensor module through the I2C interface. The slave first sends the signal to the host’s RF wireless transceiver through the RF wireless transceiver through the SPI interface, and then sends it to the host’s microcontroller, the host’s microcontroller sends signals to the smartphone via the Bluetooth module.

B. DATA MINING

1) PRE-PROCESSING

The motion signal collected by the wearable sensor device will be mixed with the noise generated by the human body’s unconscious shaking, the human body’s own gravity acceleration, and the sensor’s own offset error. Therefore, after collecting the motion signals, first, use lowpass filtering to filter out high-frequency noise in acceleration and angular velocity, and then use highpass filtering to filter out the human gravity acceleration in the lowpass filtered acceleration motion signals.

- Lowpass Filtering

The motion signals of the acceleration and angular velocity sensors are collected, and a moving average filter is used to filter high-frequency noise, where moving average filter is a lowpass filtering.

\[ y[n] = \frac{1}{N} \sum_{i=1}^{N} x[n - i] \] (1)
\( x[n] \) is the motion signal \((a_{cx}, a_{cy}, a_{cz}, \omega_{cx}, \omega_{cy}, \omega_{cz})\) that needs to be input to the digital lowpass filtering for calibration, \( y[n] \) is the signal after lowpass filtering which is \((a_{hx}, a_{hy}, a_{hz}, \omega_{hx}, \omega_{hy}, \omega_{hz})\), and \( N \) is the number of points in the average filter. In this article, after repeated tests, we set the motion signals generated by the acceleration and angular velocity sensors to \( N = 20 \) and 10.

- **Highpass Filtering**
  After lowpass filtering, we are using highpass filtering to remove gravity acceleration from accelerations. In this article, a third-order highpass filter with an intercept frequency of 0.005Hz is used to filter the acceleration of the gravity to obtain the signal of the acceleration sensor, which is \((a_{hx}, a_{hy}, a_{hz})\).

### 2) FEATURE EXTRACTION

The motion signal collected by the wearable sensor device is a form of a data stream, which is not suitable for directly extracting features. Data windowing is usually preprocessed before feature extraction of motion signals. Among them, sliding windows are widely used because they are simple, intuitive and highly real-time. They are widely used in wearable sensor network activity recognition algorithms [26], [27]. This paper uses a 50% overlap between adjacent windows. The overlapping windows have better smoothness, are more suitable for analyzing continuous data, and can meet the requirements of real-time processing. After performing the windowing process, the length of the original motion signal is unified, and there is a common standard for comparison between different motion signals. Next, we extracted as many features as possible from the time and frequency domains and extracted the most effective feature vectors from the original sensor motion signal. Because the time-domain analysis method directly extracts features from the time-domain motion signal in the sensor, compared with the other methods for extracting features, the time-domain feature extraction is simple and suitable for real-time systems. In the time domain, we calculated the mean, variance, standard deviation, maximum value, minimum value, correlation coefficient, range, and mode. Many time-domain features have clear physical meanings. Standard deviation reflects the degree of dispersion of the sensor signal, and the correlation coefficient reflects the correlation between the two coordinate axes. Then we applied the fast fourier transform algorithm to extract the features in the frequency domain. In the frequency domain, we extracted the DC components, the statistical characteristics of the amplitude and the statistical characteristics of the shape (mean, variance, standard deviation, slope, kurtosis). We extract a total of 456 \((4 \times 2 \times 3 \times 19)\) features. Due to the large range difference between the extracted different feature values, we normalize the extracted features into the interval \([0, 1]\) so that the features of different dimensions have a certain degree of comparison in value, which will greatly improve the accuracy of the classifier.

\[
y_i = \frac{s_i - s_{\min}}{s_{\max} - s_{\min}}
\]

where \( s_i \) represents the extracted feature value, \( s_{\max} \) and \( s_{\min} \) are the maximum and minimum values of the extracted feature value, respectively, and \( y_i \) represents the feature value after normalization.

### 3) FEATURE REDUCTION

After feature extraction, because there are too many extracted features, using high-dimensional features for classification directly will encounter many problems. For example, the existence of a large amount of redundancy between features will not only reduce the performance of the classifier but also increase the computational complexity, not intuitive enough, and data visibility is poor. This paper uses principal component analysis to reduce the dimension of the collected feature vectors, find the optimal linear combination of feature vectors, and rank their importance. The main idea of principal component analysis is to replace the original high-dimensional feature data with a new low-dimensional feature set so that it displays as much information as possible about the original data.

### C. TIME SERIES ANALYSIS

1) **A TEMPORAL-ATTENTION BASED ENCODER-DECODER MODEL FOR MULTIVARIATE TIME SERIES PREDICTION**

To capture the temporal characteristic of collected motion data, this paper proposes a temporal-attention based encoder-decoder model for multivariate time series prediction. The aim of this model is to make a long-term prediction with multivariate sequences, which has more practical significance for motion prediction, especially for real-time exercises guidance. The overall model framework is presented in Fig. 3. This model can input historical time series samples in the encoder, and use the time attention mechanism in the decoder to automatically select the corresponding encoder hidden state across all steps.

- **Notation**

The actual sensor data includes attributes of multiple variables, such as acceleration data \((a_{hx}, a_{hy}, a_{hz})\) and angular velocity data \((\omega_{hx}, \omega_{hy}, \omega_{hz})\), which are collectively called exogenous sequences; the modulus of acceleration data is taken as the target sequence. Given \( n \) exogenous sequences and a target sequence, we use \( \mathbf{X} = (x_1, x_2, \ldots, x_T)^T \in \mathbb{R}^{n \times T} \), where \( T \) represents the length of the window, and we use \( \hat{x}^k \in \mathbb{R}^T \) to represent the \( k \)-th exogenous sequence within the window length \( T \). For the previous target sequence, we use \( \tilde{y} = (y_1, y_2, \ldots, y_T)^T \in \mathbb{R}^T \) to represent the target sequence with window size \( T \), let \( \tilde{x} = (\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_T) \) be the input sequence of the model, and where \( \tilde{x}_i = (x^1_i, x^2_i, \ldots, x^n_i, y_i) \in \mathbb{R}^{n+1} \) to represent the \( n \) exogenous sequence and a target sequence at time \( T \).
Encoder and Decoder components

- **Problem Statement**
  We use the union \( \tilde{x}_t = [x_t; y_t] \), which exogenous sequence \((x_1, x_2, \ldots, x_T)\) and the target sequence \((y_1, y_2, \ldots, y_T)\) as input to predict the next \( N \) time steps, which can be expressed as \( \hat{Y} = \{\hat{y}_{T+1}, \hat{y}_{T+2}, \ldots, \hat{y}_{T+N}\} \) in \( \mathbb{R}^N \).

- **Encoder and Decoder**
  The encoder selects the LSTM unit to capture the dependencies of time series. Assume for the input sequence \( X = (x_1, x_2, \ldots, x_T) \), where \( x_t \in \mathbb{R}^n \), where \( n \) is the number of features of the sensor data, and the value of the previous hidden state to calculate the output of the sequence \( y_1, y_2, \ldots, y_T \), the output at \( t \) time is
  \[
  y_t = f_a(h_{t-1}, x_t)
  \]
  where \( h_{t-1} \in \mathbb{R}^m \) is the hidden state at time \( t - 1 \), \( m \) represents the size of the hidden state, and \( f_a \) is a non-linear activation function.

- **Temporal-attention based encoder and decoder mechanism**
  We propose a novel temporal-attention based encoder-decoder model for wearable multi-sensory time series prediction. The proposed model uses a recurrent neural network based on LSTM as the encoder. The encoder is essentially an RNN. Sutskever [24] used LSTM as an encoder-decoder during machine translation.

  In terms of time prediction, for a given feature sequences \( \tilde{X} = (\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_T) \), where \( \tilde{x}_t \in \mathbb{R}^{n+1} \), where \( n + 1 \) represents \( n \) exogenous sequences and a target sequence. The mapping from \( \tilde{x}_t \) to \( h_t \) is done at time \( t \) by the LSTM encoder.
  \[
  h_t = f_d(h_{t-1}, \tilde{x}_t)
  \]
  where \( h_t \in \mathbb{R}^m \), \( h_t \) is the hidden state of the encoder at time \( t \), and \( m \) is the size of the hidden state. \( f_d \) is a non-linear function. The proposed model uses the LSTM to update the hidden state of the encoder. The hidden state \( h_T \) of the encoder at time \( T \) is used as the initial hidden state of the decoder. The product of \( h_T \) and a fully connected matrix is used as the initial prediction value of the decoder. Since the above two calculations are very simple, so don’t write the relevant formula here.

  As Cho et al. [25] introduced in the machine translation, usually the encoder-decoder model encodes the input sequence into a fixed-length vector. The decoder generates a predicted time series based on the vector, but as the length of the input sequence increases, the performance of the encoder-decoder model will continue to decrease. Using the attention mechanism in the decoder allows the hidden state of the encoder and the target sequence to be automatically aligned. The calculation of the weight of attention is as follows.

  \[
  e_i^t = V_c \tanh(W_c [h_{t-1}; s_{t-1}])
  \]
  \[
  \alpha_i^t = \frac{\exp(e_i^t)}{\sum_{j=1}^{T} \exp(e_j^t)}
  \]
  Among them, \( V_c \in \mathbb{R}^m \) and \( W_c \in \mathbb{R}^{m \times 2m} \) are parameters to be learned, and \( h_{t-1} \in \mathbb{R}^m \) and \( s_{t-1} \in \mathbb{R}^m \) are the hidden state of the encoder and the hidden state of the decoder, respectively, \( m \) is the size of the hidden state. Where attention weight \( \alpha_i^t \) indicates the importance of the \( i \)th encoder’s hidden state to prediction at time \( t \). Then, using the attention weight \( \alpha_i^t \) and the encoder’s hidden state \( \{h_1, h_2, \ldots, h_T\} \) to perform weighted summation to obtain the context vector \( c_t \), which can be calculated by the following formula:

  \[
  c_t = \sum_{i=1}^{T} \alpha_i^t h_i
  \]
  The next step is to calculate the hidden state of the decoder at the current time \( t \) through the combination of the context vector \( c_t \) and the predicted value \( \hat{y}_{T+t-1} \)
at time \( t - 1 \), plus the previously hidden state \( s_{t-1} \) of the decoder.

\[
s_t = f_a \left( \left[ \hat{y}_{T+t-1}; c_t \right], s_{t-1} \right)
\]

where \( \hat{y}_{T+t-1}; c_t \in R^{m+1} \), \( s_{t-1} \in R^m \), \( f_a \) are non-linear functions, the proposed model uses LSTM as the update of the hidden state \( s_t \) of the decoder.

In the last step, the predicted value \( \hat{y}_{T+t} \) at the moment \( t = 1 \), the context vector \( c_t \) and the hidden state \( s_t \) of the decoder at the current moment obtained in the above formula are re-connected in series, and then the final predicted value is obtained through two full connection matrices.

\[
\hat{y}_{T+t} = W_y U_y \left( \left[ \hat{y}_{T+t-1}; c_t; s_t \right] \right)
\]

where \( U_y \in R^{(2m+1) \times m} \) and \( W_y \in R^m \) are the parameters which need to be learned.

2) A LSTM BASED ACTIVITY RECOGNITION ALGORITHM

To get the optimized classification results of human daily activities, a LSTM based activity recognition algorithm is proposed in this framework. The block diagram of the activity recognition algorithm is shown in Fig. 4. The daily signals of human beings are identified through the motion signals collected by the wearable sensor device. The collected data samples are preprocessed, feature constructed, and finally the LSTM classifier recognizes daily activities. The following briefly introduces the classifier algorithm.

The LSTM is composed of three gates: forget gate \( f_t \), input gate \( i_t \), and output gate \( o_t \). The forget gate determines what is discarded from the cell state, and the input gate determines how much new information is entered into the cell state in the next step. The output gate outputs those values based on the state of the cell.

\[
f_t = \sigma(U_f x_t + W_f h_{t-1} + b_f)
\]

\[
i_t = \sigma(U_i x_t + W_i h_{t-1} + b_i)
\]

\[
o_t = \sigma(U_o x_t + W_o h_{t-1} + b_o)
\]

\[
s_t = f_t \odot s_{t-1} + i_t \odot \tanh(U_s h_{t-1} + W_s x_t + b_s)
\]

\[
h_t = o_t \odot \tanh(s_t)
\]

where \( \sigma \) represents the logistic sigmoid function, \( \tanh \) represents the hyperbolic activation function, \( \odot \) represents element-wise multiplication.

IV. EXPERIMENTS

In this section, we run extensive experiments to verify the proposed algorithm. For motion long-term prediction, 10 participants carried out squat motion series for 10 times. Each time includes 5 squat movements. The sample rate from the proposed prototype is 50 samples per second. In total, we collected 81536 valid data sample to verify the proposed algorithm.

A. MULTI-TAED

1) DATASET

We carried out extensive experiments to verify the proposed algorithms. For motion long-term prediction, 10 participants carried out squat motion series for 10 times. Each time includes 5 squat movements. The sample rate from the proposed prototype is 50 samples per second. In total, we collected 81536 valid data sample to verify the proposed algorithm.

2) BASELINE METHOD

We use 5 models to compare with proposed Multi-TAED methods:

**ARIMA** [16]: Used to predict future values in a time series. It is widely used in the field of time series.

**SVR** [18]: SVR is the application of support vector in the field of regression functions, and it performs well in time series prediction.

**LSTM** [20]: Paper was first published in 1997. Because the unique design structure can solve the vanishing gradient problem in RNN, LSTM is suitable for processing and predicting important events with very long intervals and delays in time series.

**GRU** [21]: GRU is a variant of LSTM. Although LSTM has many variants, GRU maintains the effect of LSTM while making the structure simpler. Similarly, GRU also solves the vanishing gradient problem of RNN, and it is also applicable to time series.

**SEQ2SEQ** [24]: SEQ2SEQ is a model of the encoder-decoder structure. The encoder encodes a variable-length time series into a fixed-length vector, and the decoder performs iterative prediction.

3) PARAMETER SETTING AND EVALUATION METRICS

Most of the previous works used intelligent algorithms to predict short-term changes. The application scenario of proposed Multi-TAED is to predict human motion sequence for a period of time in the future, so it is a long-term prediction.

We run a group of experiments on different future time step \( N \in \{8, 16, 32, 64, 128\} \). During training, the batch size is set to 256, and the learning rate is set to 0.001. The two hyperparameters in the model are the length of the time window and the size of the hidden state of the encoder and decoder. For the length of the time window \( T \). We set it to \( T \in \{8, 16, 32, 64, 128\} \). For a given future time step \( N \), we need to perform a grid search on a set time window \( T \).
to select the best time window parameters for the prediction of each future time step. For the size of the hidden state, most previous works used 64, 128 or 256. In the experiment, we compared the sizes of the three different hidden states, and we found that the experiment with the hidden state size of 128 works best, so in this article, we set it to \( m = 128 \). In order to ensure the fairness of the experiment, these parameters are also applicable to SEQ2SEQ, GRU, LSTM for the baseline methods.

To evaluate the performance of the proposed algorithm, root mean squared error (RMSE), mean absolute error (MAE), and mean absolute percentage error (MAPE) are utilized. Assuming \( y \) is the true value of the experiment, and \( \hat{y} \) is the predicted value at time window \( T \). The definition of the evaluation function is as follows:

\[
RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (\hat{y}_i - y_i)^2} \quad \text{(15)}
\]

\[
MAE = \frac{1}{N} \sum_{i=1}^{N} |\hat{y}_i - y_i| \quad \text{(16)}
\]

\[
MAPE = \frac{1}{N} \sum_{i=1}^{N} \left| \frac{\hat{y}_i - y_i}{y_i} \right| \times 100\% \quad \text{(17)}
\]

4) RESULTS AND DISCUSSION

We compared the prediction performance of the proposed algorithm with the other five baselines, which includes two classic machine learning models (ARIMA, SVR) and three latest deep learning models (LSTM, GRU, SEQ2SEQ). We set the time window \( T \in \{8, 16, 32, 64, 128\} \) of the input, and select a time window with the best effect as the input through the grid search. When we change the time step \( N \) of the output, other parameters remain the same. Table 1 shows the experimental results of six models. Our model outperforms other models in predicting the performance of multi-step time series. When the short-term prediction length is set to 16, the evaluation indexes MAP, MAPE, and RMSE are 0.2169, 2.3551, and 0.2921, respectively. When we set the long-term forecast length to 64, the three indicators of the evaluation model are also the lowest, which are 0.4999, 5.4600, and 0.7192. This shows that our attention-based encoder-decoder model improves prediction performance. In addition, the predictive performance of deep learning models in multi-step time series is better than classic machine learning models. For example, the performance of ARIMA model is much worse than other methods, this model requires that the target sequence is not only stationary but also has a linear relationship, so this model has considerable limitations in time series prediction. Kernel-based SVRs do not capture the correlation between time series. Models based on a single LSTM unit or GRU unit can capture time dependence, but the effects of the dynamic characteristics of the exogenous sequence on the target sequence are not considered. Sequence-to-sequence encoder-decoder learning models usually have better performance than recurrent neural networks (LSTM, GRU). However, as the length of the encoder increases, a fixed-length vector encoded is not sufficient to represent the time-dependent characteristics between sequences. This shows that in some cases, attention-based encoder-decoder learning models are more effective in time series prediction than classic machine learning models and deep learning models. In addition, we found that the choice of the size of the prediction length \( N \) has a great impact on the prediction performance. As shown in Fig. 5, as the time step \( N \) of the prediction increases, the accuracy of the model’s prediction results will continue to decline, which is also in line with our prediction. The longer the future, the worse the prediction effect of the model. But we can observe that our model has the best prediction performance under different prediction time steps. So compared with the classic learning model and the deep learning model, the long-term prediction model of the multivariate time series we proposed has better performance in both the shorter and longer prediction lengths. This means that our proposed temporal-attention based encoder-decoder model for wearable multi-sensory time series can well learn the relationships and long-term dependencies between multiple time series features. In order to compare the prediction performance of each model, we performed a visualization. We show the prediction results of the four models and the true value of the data set in Fig. 6. All models are compared in the case of predicting 32-time steps, it can be found from the figure that the prediction results of our model are closest to the real situation. To make the data visualization clearer, we only show a comparison of 400 predicted and actual values. In addition, since the predicted values of the ARIMA and SVR models in the long-term prediction have large errors relative to the true values, they are omitted in Fig. 5 and Fig. 6.
randomly select some samples and features. Bagging Tree method is that the Random Forest will with the largest class interval from it.

ination ability for classification, and constructs a classifier matically finds out which support vectors have better discrim-

structural risk minimization, the support vector machine auto-
catory, the sample is also divided into this category.

bors of a sample in the feature space belong to a certain classifiers, specifying the category with the most votes as the difficult to predict for continuous fields or time series.

the output of attribute features, and each leaf node represents the final classification result. But Decision Trees are more accuracy of the classification.

We use 6 baseline models to compare with our model.

Naive Bayes [28]: First calculate the probability density function of each feature, and then calculate the posterior probability that the sample to be tested belongs to different categories. Because there is a relationship between the attributes of most data sets, this problem will lead to a reduction in the accuracy of the classification.

Decision Tree [29]: The process of Decision Tree classification starts from the root node, each branch represents the output of attribute features, and each leaf node represents the final classification result. But Decision Trees are more difficult to predict for continuous fields or time series.

Bagging Tree [30]: Bagging Tree is a classifier with multiple Decision Trees, and integrates the voting results of all classifiers, specifying the category with the most votes as the final output.

K-NearestNeighbor [31]: If most of the k nearest neighbors of a sample in the feature space belong to a certain category, the sample is also divided into this category.

Support Vector Machine [32]: Based on the theory of structural risk minimization, the support vector machine automatically finds out which support vectors have better discrimination ability for classification, and constructs a classifier with the largest class interval from it.

Random Forest [33]: The only difference from the above Bagging Tree method is that the the Random Forest will randomly select some samples and features.

3) EVALUATION METRICS

We use four metrics to evaluate the proposed algorithm. They are accuracy (Acc), precise (Pre), recall (Re) and F1(F1 score). True positive (TP): classify the samples that belong to the positive class into positive class; true negative (TN): classify samples that originally belonged to the negative class into negative classes; false positive (FP): classify errors that originally belonged to the negative class into positive classes; false negative (FN): classify errors that were originally positive into negative. F1 score can be regarded as a harmonious average of model accuracy and recall. In this paper, $\beta = 1$. The formula is defined as follows:

$$
\text{Acc}(\%) = \frac{TP + TN}{TP + TN + FP + FN} \times 100
$$

(18)

$$
\text{Pre}(\%) = \frac{TP}{TP + FN} \times 100
$$

(19)

$$
\text{Re}(\%) = \frac{TP + FP}{TP + FN} \times 100
$$

(20)

$$
F1 - \text{score} (\%) = (1 + \beta^2) \times \frac{\text{Pre} \times \text{Re}}{(\beta^2 \times \text{Pre}) + \text{Re}} \times 100
$$

(21)

4) RESULTS AND DISCUSSION

We collected six types of human movements through the wearable sensor network, including 1) squat (S1), 2) raise leg (S2), 3) go up the stairs (S3), 4) go down the stairs (S4), 5) walk (S5), 6) open and close jump (S6). The first two actions are collected in the laboratory, and the other four actions are collected in the stadium. We used a deep learning classifier LSTM and six different activity recognition algorithms to make comparisons in order to ensure the fairness of the experiments. Table 2 presents a comparison of the recognition performance of the activity recognition algorithm in 5-fold and 10-fold cross-validation strategies. Through ten-fold cross-validation, the F1-score of Naive Bayes, Decision Tree, Bagging Tree, Support Vector Machine, K-NearestNeighbor, and Random Forest in the baseline were 90.61%, 94.78%, 95.79%, 95.94%, 95.87%, 96.38%, and the F1-score of our model reached 99.26%. In addition, the classification performance of our model is still the best in five-fold cross-validation, of which F1-score is 98.89%. The classification performance of the classic machine learning classifiers is also very different. Among them, the performance of Naive
TABLE 2. Comparison of the recognition performance of LSTM classifier and 6 baseline activity recognition algorithm classifiers in 5-fold and 10-fold cross-validation strategies.

| Models               | 5-fold | 10-fold |
|----------------------|--------|---------|
|                      | Acc(%) | Pre(%)  | Re(%) | F1(%) | Acc(%) | Pre(%)  | Re(%) | F1(%) |
| Naive Bayes          | 90.80  | 90.67   | 90.55 | 90.57 | 90.83  | 90.76   | 90.58 | 90.61 |
| Decision Tree        | 94.65  | 94.67   | 94.68 | 94.66 | 94.81  | 94.80   | 94.80 | 94.78 |
| Bagging Tree         | 95.44  | 95.69   | 95.42 | 95.52 | 95.77  | 95.89   | 95.75 | 95.79 |
| Support Vector Machine | 95.68  | 96.79   | 95.35 | 95.84 | 95.77  | 96.81   | 95.46 | 95.92 |
| K-NearestNeighbor    | 95.78  | 96.04   | 95.63 | 95.71 | 95.91  | 96.17   | 95.80 | 95.87 |
| Random Forest        | 95.99  | 96.68   | 95.83 | 96.13 | 96.23  | 96.86   | 96.13 | 96.38 |
| LSTM                 | 98.97  | 98.91   | 98.87 | 98.89 | 99.27  | 99.26   | 99.28 | 99.26 |

Bayes classification is the worst; Decision Trees, Bagging Trees, Support Vector Machines, K-NearestNeighbor, and Random Forest classifiers have little difference in accuracy. However, in the five-fold and ten-fold cross-validation, our model based on F1-score index evaluation still has a nearly 3% improvement over the best model Random Forest in the baseline. This shows that our deep learning classifier LSTM model improves the performance of classification. Table 3 shows the confusion matrix for daily activities with ten-fold cross-validation using deep learning LSTM. The matrix not only lists the recognition accuracy of various actions but also shows the degree of confusion between different actions. We also found that the errors mainly come from going upstairs, going downstairs, and walking; the reason is that the three active motion signals are very similar. Squats and open and close jump have the highest accuracy due to the large differences between the movements.

V. CONCLUSION

In this paper, we propose an attention-based encoder-decoder framework for multi-sensory time-series analytic of wearable sensor devices. It integrates a multivariate time series long-term prediction algorithm and an activity recognition algorithm based on deep learning method. Through a large number of experiments, the proposed Multi-TAED model has achieved acceptable results in both short-term and long-term predictions. It has been verified that the attention mechanism can capture the relationship between multivariate time series and achieve good prediction results. The RMSE errors generated by prediction at 16 and 64 time steps are 0.2921 and 0.7192, respectively. In the activity recognition algorithm, the proposed framework adopts the deep learning model LSTM as the classifier for the recognition algorithm to identify human daily activities and sports. The results have been extensively verified by the 5-fold and 10-fold cross-validation strategies. For six different activities, the performance of the proposed algorithm in terms of F1-score achieved 99.26% and 98.89%, respectively, which outperforms other machine-learning-based algorithms. In the future, we will investigate how to deploy a light-weight model on resource-constrained hardware and get acceptable results as well as reducing the communication cost.

REFERENCES

[1] S. Du, T. Li, X. Gong, Z. Yu, Y. Huang, and S.-J. Horng, “A hybrid method for traffic flow forecasting using multimodal deep learning,” 2018, arXiv:1803.02099. [Online]. Available: https://arxiv.org/abs/1803.02099
[2] H. Li, Y. Shen, and Y. Zhu, “Stock price prediction using attention-based multi-input LSTM,” in Proc. Asian Conf. Mach. Learn., 2018, pp. 454–469.
[3] M. Cornacchia, K. Ozcan, Y. Zheng, and S. Velipasalar, “A survey on activity detection and classification using wearable sensors,” IEEE Sensors J., vol. 17, no. 2, pp. 386–403, Jan. 2017.
[4] J. Wu, S. Pan, X. Zha, C. Zhang, and X. Wu, “Multi-instance learning with discriminative bag mapping,” IEEE Trans. Knowl. Data Eng., vol. 30, no. 6, pp. 1065–1080, Jun. 2018.
[5] J. Margarito, R. Helou, F. Bianchi, F. Sartor, and A. Bonomi, “User-independent recognition of sports activities from a single wrist-worn accelerometer: A template matching based approach,” IEEE Trans. Biomed. Eng., vol. 63, no. 4, pp. 788–796, Aug. 2015.
[6] W.-X. Lu, C. Zhou, and J. Wu, “Big social network influence maximization via recursively estimating influence spread,” Knowl.-Based Syst., vol. 113, pp. 143–154, Dec. 2016.
[7] L. Gao, J. Wu, C. Zhou, and Y. Hu, “Collaborative dynamic sparse topic regression with user profile evolution for item recommendation,” in Proc. 31st AAAI Conf. Artif. Intell., 2017.
[8] Y. Qin, D. Song, H. Chen, W. Cheng, G. Jiang, and G. Cottrell, “A dual-stage attention-based recurrent neural network for time series prediction,” 2017, arXiv:1704.02971. [Online]. Available: https://arxiv.org/abs/1704.02971
[9] Y.-P. Chen, J.-Y. Yang, S.-N. Liou, G.-Y. Lee, and J.-S. Wang, “Online classifier construction algorithm for human activity detection using a triaxial accelerometer,” Appl. Math. Comput., vol. 205, no. 2, pp. 849–860, Nov. 2008.

[10] K. Altun, B. Barshan, and O. Tunçel, “Comparative study on classifying human activities with miniature inertial and magnetic sensors,” Pattern Recognit., vol. 43, no. 10, pp. 3605–3620, Oct. 2010.

[11] L. Bao and S. S. Intille, “Activity recognition from user-annotated accelerometer data,” in Proc. Int. Conf. Pervasive Comput. Springer, 2004, pp. 1–17.

[12] A. M. Khan, Y.-K. Lee, S. Y. Lee, and T.-S. Kim, “A triaxial accelerometer-based physical-activity recognition via augmented-signal features and a hierarchical recognizer,” IEEE Trans. Inf. Technol. Biomed., vol. 14, no. 5, pp. 1166–1172, Sep. 2010.

[13] N. Ravi, N. Dandekar, P. Mysore, and M. L. Littman, “Activity recognition from accelerometer data,” in Proc. AAAI, vol. 5, 2005, pp. 1541–1546.

[14] A. Anjum and M. U. Ilyas, “Activity recognition using smartphone sensors,” in Proc. IEEE 10th Consum. Commun. Netw. Conf. (CCNC), Jan. 2013, pp. 914–919.

[15] N. Nair, C. Thomas, and D. B. Jayagopi, “Human activity recognition using temporal convolutional network,” in Proc. 5th Int. Workshop Sensor-Based Activity Recognit. Interact., 2018, p. 17.

[16] G. E. P. Box and D. A. Pierce, “Distribution of residual autocorrelations in autoregressive-integrated moving average time series models,” J. Amer. Stat. Assoc., vol. 65, no. 332, pp. 1509–1526, Dec. 1970.

[17] S. Chen, X. X. Wang, and C. J. Harris, “NARX-based nonlinear system identification using orthogonal least squares basis hunting,” IEEE Trans. Control Syst. Technol., vol. 16, no. 1, pp. 78–84, Jan. 2008.

[18] T. Van Gestel, J. Suykens, D.-E. Baestaens, A. Lambrechts, G. Lanckriet, B. Vandaele, B. De Moor, and J. Vandewalle, “Financial time series prediction using least squares support vector machines within the evidence framework,” IEEE Trans. Neural Netw., vol. 12, no. 4, pp. 809–821, Jul. 2001.

[19] J. L. Elman, “Distributed representations, simple recurrent networks, and grammatical structure,” Mach. Learn., vol. 7, nos. 2–3, pp. 195–225, 1991.

[20] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” Neural Comput., vol. 9, no. 8, pp. 1735–1780, 1997.

[21] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, “Empirical evaluation of gated recurrent neural networks on sequence modeling,” 2014, arXiv:1412.3555. [Online]. Available: https://arxiv.org/abs/1412.3555.

[22] S. Xingjian, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-C. Woo, “Convolutional LSTM network: A machine learning approach for precipitation nowcasting,” in Proc. Adv. Neural Inf. Process. Syst., 2015, pp. 802–810.

[23] S. C. Prasad and P. Prasad, “Deep recurrent neural networks for time series prediction,” 2014, arXiv:1407.5949. [Online]. Available: https://arxiv.org/abs/1407.5949.

[24] I. Sutskever, O. Vinyals, and Q. Le, “Sequence to sequence learning with neural networks,” Adv. NIPS, 2014.

[25] K. Cho, B. Van Merriënboer, D. Bahdanau, and Y. Bengio, “On the properties of neural machine translation: Encoder-decoder approaches,” 2014, arXiv:1409.1259. [Online]. Available: https://arxiv.org/abs/1409.1259.

[26] N. C. Krishnan and D. J. Cook, “Activity recognition on streaming sensor data,” Pervasive Mobile Comput., vol. 10, pp. 138–154, Feb. 2014.

[27] J. R. Kwapisz, G. M. Weiss, and S. A. Moore, “Activity recognition using cell phone accelerometers,” ACM SigKDD Explor. Newslett., vol. 12, no. 2, pp. 74–82, 2011.

[28] O. D. Lara, A. J. Pérez, M. A. Labrador, and J. D. Posada, “Centinela: A human activity recognition system based on acceleration and vital sign data,” Pervasive Mobile Comput., vol. 8, no. 5, pp. 717–729, 2012.

[29] M. Ernes, J. Parkka, J. Mantyjarvi, and I. Korhonen, “Detection of daily activities and sports with wearable sensors in controlled and uncontrolled conditions,” IEEE Trans. Inf. Technol. Biomed., vol. 12, no. 1, pp. 20–26, Jan. 2008.

[30] L. Breiman, “Bagging predictors,” Mach. Learn., vol. 24, no. 2, pp. 123–140, 1996.

[31] S. Preece, J. Goulermas, L. Kenney, and D. Howard, “A comparison of feature extraction methods for the classification of dynamic activities from accelerometer data,” IEEE Trans. Biomed. Eng., vol. 56, no. 3, pp. 871–879, Mar. 2009.

[32] A. Krause, M. Ihmig, E. Rankin, D. Leong, S. Gupta, D. Siewiorek, A. Smailagic, M. Deisher, and U. Sengupta, “Trading off prediction accuracy and power consumption for context-aware wearable computing,” in Proc. 9th IEEE Int. Symp. Wearable Comput. (ISWC), Dec. 2005, pp. 20–26.

[33] M. Pal, “Random forest classifier for remote sensing classification,” Int. J. Remote Sens., vol. 26, no. 1, pp. 217–222, Jan. 2005.