LSTM Step Prediction and Ontology-Based Recommendation Generation in Activity eCoaching

Ayan Chatterjee
Department of Information Technology
Center for eHealth, University of Agder
Grimstad, Norway
ayan.chatterjee@uia.no

Nibedita Pahari
Department of Software Engineering
Software Engineering and IT Services, Knowit As
Grimstad, Norway
nibedita.pahari@knowit.no

Michael Riegler
Department of Holistic Systems
Simula Metropolitan Center for Digital Engineering (SimulaMet)
Oslo, Norway
michael@simula.no

Andreas Prinz
Department of Information Technology
Center for eHealth, University of Agder
Grimstad, Norway
andreas.prinz@uia.no

***Abstract*—An eCoach system may allow people to manage a healthy lifestyle with health state monitoring (e.g., physical activity) and personalized recommendation generation. Daily step count is an important feature to provide a direct and indirect reflection on individual activity levels. Therefore, a personalized, predictive model may be beneficial to forecast future "steps" to motivate participants based on the temporal "step" pattern. Here, we have conceptualized the idea with a Bidirectional Long-Short-Term-Memory (LSTM) model for weekly activity forecasting and a rule-base for personalized recommendation generation with Ontology reasoning and querying in activity eCoaching. First, we have used the publicly available "PMData" dataset of 16 adults (M: 13; F:3) to train and test the models and explore the possibility of accurate univariate time-series forecasting of "step counts". Second, we have created an Ontology and a rule-base to generate personalized activity recommendations to motivate participants to accomplish their activity goals (e.g., complete "X" steps daily and stay active for the entire week).**

Keywords— Activity Prediction; LSTM; eCoach; Ontology; Recommendation Generation.

I. INTRODUCTION

Regular physical activity has a positive influence on the prevention of lifestyle diseases. Compared with people with sufficient exercise, people with insufficient activity have an increased risk of death by 20% to 30% [1][2]. Tudor Locke et al. [3] and Matthews et al. [4] showed that people's exercise differs between weekends and weekdays regardless of gender. Gardner et al. [4] stated that self-monitoring, problem-solving, and reforming the social or physical environment are the most promising strategies for behavior change. Intervention design to increase physical activity levels and reduce sedentary time varies significantly in content and usefulness [5-9]. The use of mobile applications (apps.) appears to increase physical activity; however, the research results in discrepancies to reach the claimed outcomes [10][11]. Mobile apps used to improve physical activity levels should include a personalized feedback loop and provide appropriate assistance [12]. Only a few such mobile apps have been reviewed, and the evidence is of poor quality [12].

An activity tracker is maintained in physical activity monitoring to note down daily step count, metabolic equivalent of tasks, kilocalories, and total distance traveled. Such data are recorded over time with a data collection module and analyzed by a decision module to give personalized feedback to achieve individual activity goals. Also, the decision module suggests changing individual behavior, daily routine, and activity plan [5]. The pedometer provides an objective measurement of activity level and enables self-monitoring. Furthermore, most modern consumer-based activity trackers already contain a variety of behavior change models or theories [13][14]. A meta-analysis from Qiu et al. [15] and Stephenson et al. [16] concluded that using a pedometer has a small but significant effect on reducing sedentary time. Just wearing an activity tracker can stimulate the passion for performing activities to improve the quality of life. Altogether, reducing the sedentary time by increasing physical activity necessitates self-motivation, self-correlation, and self-management. A human coaching process may improve self-management of personal activities toward a healthy lifestyle. Several studies have been conducted on health coaching to generate recommendations for healthy lifestyles. The idea is to provide remote care and advice for fitness programs to prevent lifestyle diseases. Coaching processes can be either "face-to-face" or "technology-driven" (via telematic means) [17][18]. The face-to-face coaching with manual activity tracking and recommendation generation is time-consuming and tiresome.

Therefore, an eCoach-based continuous health monitoring and recommendation generation have become popular to provide remote and timely assistance to its participants with the advancement of Information and Communication Technologies (ICTs). An eCoach system can generate automatic, evidence-based, and personalized lifestyle recommendations to achieve personal lifestyle goals. Real-time analysis of data and, thereby, the generation of timely tailored recommendations are vital in eCoaching. From the literature search, the eCoach concept in healthcare is still in the nascent stage, and there is very little research conducted on actual sensor data using machine learning technology.

An activity eCoach system may generate automatic and convenient tailored recommendations based on the insights from activity data (collected through a wearable Bluetooth activity device such as Fitbit, Actigraph, MOX2-5) and personal preference data to accomplish personal activity goals. Different research has studied sensor data to classify or forecast human activities [19-21]. However, only a few studies have investigated the use of actionable and data-driven predictive

978-1-6654-6975-3/22/$31.00 ©2022 IEEE

models [22]. Improving individual physical activity combined with wearable activity sensors and digital activity trackers, eCoach features can be encouraging; however, missing in the existing literature.

To prove the eCoach-based personalized recommendation generation concept, we conceptualize the design of an activity eCoach prototype system that can collect personal preference data and activity sensor data from participants and store them in an Ontology with semantic annotation; process collected activity data with the LSTM algorithm [23] to forecast individual step counts and generate rule-based personalized recommendations with Ontology [24] to guide individuals to accomplish their activity goals. Based on the preferences, the goals can be daily, weekly, or monthly. The semantic annotation of activity and preference data, and recommendation messages are not in the focus of this study. The research question for this study is –

How do LSTM "step" predictions and Ontology generate tailored recommendations in activity eCoaching?

The remainder of the paper is structured as follows. Section II presents the design of the eCoach prototype system. In Section III, we describe the proposed work. In section IV, we present the adopted methods. In Section V, we discuss the experimental results, and the paper is concluded in Section VI.

II. Design of the eCoach Prototype System

Our eCoach prototype system (see Figure 1) consists of the following four processes – data collection, data processing, recommendation generation, and recommendation delivery.

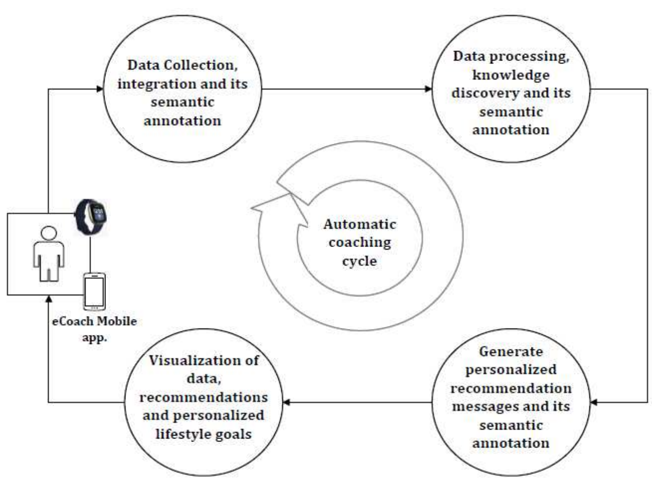


Fig. 1. The data flow in our designed eCoach prototype system.

The data collection module aims to collect preference data through questionnaires and activity data from wearable activity sensors. The data processing module forecasts the step data for the next 7-days, based on the temporal pattern in the data. The preference data consists of goal settings (e.g., daily, weekly, or monthly), target goal and steps (e.g., medium active or vigorous active and corresponding step count based on the goal settings), target score, mode of recommendation delivery (e.g., text, audio, or graph), and time of the recommendation. Participants can customize their preference data, over time. The daily step count helps to classify daily activity types as described in Table I. All the collected and predicted personal data are semantically annotated in an Ontology model written in the Web Ontology Language (OWL). In the recommendation generation module, the Ontology helps to generate rule-based recommendations with SPARQL (SPARQL Protocol and RDF Query Language) query engine. In the created rule base, logical rules comprise of propositional variables with (IMPLIES), (NOT), (AND), and (OR) operations. The rules are of two types: related to activity, and satisfiability. Satisfiability ensures that only one message will be triggered at a time. The recommendation messages are formal and informal, and their delivery depends on preferences.

TABLE I
THE RULE FOR STEP-BASED ACTIVITY CLASSIFICATION

Active class	Rule	Score
Sedentary (S)	$\sum$ steps $\leq$ 4999	0
Low active (LA)	5000 $\leq \sum$ steps $\leq$ 7499	1
Active (A)	7500 $\leq \sum$ steps $\leq$ 9999	2
Medium active (MA)	10000 $\leq \sum$ steps $\leq$ 12499	3
Highly active (HA)	$\sum$ steps $\geq$ 12500	4

III. Proposed Work

A. Ontology Modeling

The idea of ontology was created thousands of years ago in the philosophical domain. It has the design flexibility of using existing ontology to solve real-world modeling and knowledge representation problems. It supports an open-world assumption (OWA) knowledge representation style with the following elements: classes, individuals/objects, attributes, relationships, and axioms [24]. An ontology follows a connected, acyclic, and directed parsing tree structure [25]. Our Ontology has been explained in Textbox 1 and depicted in Figure 2. The purpose of the proposed Ontology is the semantic representation of the knowledge, reasoning, and rule-based decision making with the generalization rules in the induction phase.

TEXTBOX 1. THE ONTOLOGY STRUCTURE AND KNOWLEDGE EXPRESSION.

An ontology can be defined as a tuple $\Omega = \{\acute{C}, R\}$, where $\acute{C}$ is the set of concepts and R is a set of relations [25].

L = Levels (O_h) = Total number of levels in the ontology hierarchy, $0 \leq n \leq L$, where $n \in Z^+$ and n=0 represents the root node.
$C_{n,j}$ = a model classifying O at a level n; where, $j \in \{0, 1, \ldots, |C_n|\}$
$|C|$ = Number of instances classified as class C
E = Edge ($C_{n,j}$, $C_{n-1,k}$) = edge between node $C_{n,j}$ and its parent node $C_{n-1,k}$

We have used the concept and represented our ontology with four tuples:
O = {C_a, R, I, A}

C_a: {C_{a1}, C_{a2}, ….., C_{an}} represents "n" concepts or classes and each C_{ai} has a set of "j" attributes or properties $A_i = \{a_1, a_2, \ldots, a_j\}$ provided n, i, j $\in Z^+$.
R: A set of binary relations between the elements of Ca. It holds two subsets –
a. H: Inheritance relationship among concepts
b. S: Semantic relationship between concepts with a domain and range
I: Represents a knowledge base with set of object instances.
A: Represents a set of axioms to model O. A includes domain specific constraints to model an Ontology with C_a, R, and I.

The knowledge (K) in the ontology has been expressed with two tuples: K = {$Onto_{ActivityReco}$, $Rules_{ActivityReco}$},

The elements of OntoActivityReco and RulesActivityReco are:
$Onto_{ActivityReco} = \{K_L, K_B, K_C, K_D\}$
$Rules_{ActivityReco} = \{R_L, R_B, R_C, R_D\}$

K_L, K_B, K_C, K_D are the knowledge bases of the personalized physical activity recommendation's lexicon or abstraction, abduction, deduction, and induction interfaces. In contrast, R_L, R_B, R_C, R_D are set of rules to match with the abstraction, abduction, deduction, and induction interfaces, respectively. K_B, K_C, and K_D are representations of properties A of concepts (C), data or entities (e.g., activity variables), and they follow a simple representation of A(X|Y) or A(Y|X) based on the relational mapping; where, A: Attributes or properties in O, X, Y: Elements of activity variables.

All the rule execution internally follows a binary tree structure where the non-leaf nodes hold the semantic rules (A | A → B) to be executed, the leaf nodes hold the results (B or recommendation messages), and the edges hold a decision statement (True or False). Rulesets help to explain the logic behind a recommendation generation.

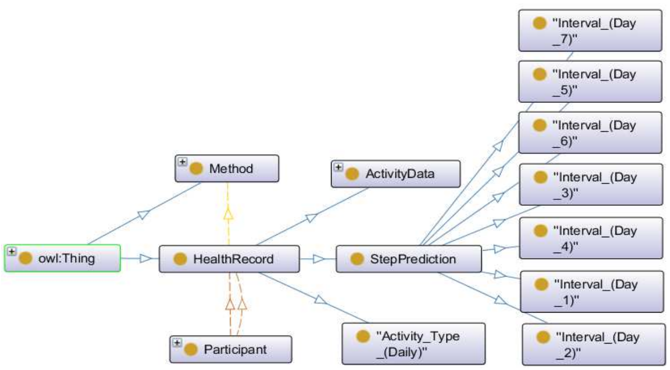


Fig. 2. The high-level structure of the proposed Ontology.

A set of propositional variables and logics, constants, and operators (such as NOT, AND, OR, IMPLIES, EQUIV, and quantifiers) are associated with Ontology representation and processing. In this study, the recommendation generation aims to maximize weekly individual physical activity to minimize sedentary time. The maximization problem to stay medium activate for a week ((Σ days (1⋯.7))) has been expressed in Textbox 2.

TEXTBOX 2. EXPRESSION FOR THE MAXIMIZATION PROBLEM.

Maximize,
$\Sigma\ Steps_{daily} \geq 70000$
$\Sigma\ GoalScore_{daily} \geq 21$
Subject to,
$Steps_{daily} \geq 10000$
$GoalScore_{daily} \geq 3$
$C_V \rightarrow P$
$P \rightarrow R$
$\Sigma\ P = 1$
Where,
V = A set of activity variables = $\{V_1, V_2, \ldots, V_n \mid n \in Z+\}$
P = A set of propositional variables = $\{P_1, P_2, \ldots, P_n \mid n \in Z+\}$
C_V = Set of activity variable combinations to create semantic rules
R = A set of recommendations = $\{R_1, R_2, \ldots, R_n \mid R_X \cap R_y = \{\emptyset\}, n, x, y \in Z+\}$
$\Sigma\ P = 1$ ensures satisfiability.

B. Personalized Recommendation Generation

With constant monitoring and tailored recommendation generation, the eCoach may motivate participants to reach their goals of activity management to maintain a healthy lifestyle. For example, to conceptualize the recommendation generation in our eCoach prototype system, we have considered a personal preferences table (see Table II). To determine the weekly score of personal goal achievement, we summed up the daily activity score (see Table I). eCoaching targets score maximization with constant activity monitoring and recommendation generation.

TABLE II
A SAMPLE PERSONAL GOAL PREFERENCES TABLE.

Key	Value
Goal setting	Weekly score
Target goal and steps	Stay medium active for a week (e.g., 70,000 - 87,500 steps)
Target score	21 for a week
Mode of recommendation	Text
Time of recommendation	8:00 am (Daily)

Models trained with individual step data are disjoint from the trained models for other participants. The Pseudo-code for the activity recommendation generation is described below:

Pseudocode: Personalized recommendation generation in activity eCoaching.

Input Daily individual daily activity step data (D(t)), Recommendation message table (RMT), Preference data (PD), Ontology (O).
Output: Personalized Recommendation list (RL)

1: Load previous day's activity data (S or (D(1),…..D(t-1)))
2. Pre-process S and split S into training and testing (Train_test_set)
3. Select the best configuration for the time-series forecast model (F) with the train_test_set and save F,
4. Predict future steps for the next 7 days (FS) and store in O.
5. Classify future activity type with rules in Table I on FS and store in O.
6. Assign activity score to activity types and update the weekly score for each Individuals and store in O.
7. Save the Ontology in database after consistency checking with a reasoner.
8. Execute SPARQL queries on O to determine potential activity goal achievement based on the PD.
9. Generate personalized recommendation messages based on the SPARQL query results.
10. Use PD for the delivery of RL to individuals.
11. Repeat Step 1 – Step 10 daily until the end date is reached.

IV. METHOD

A. Data Selection

Real-time collection of health data and its governance requires participant's consent. Therefore, we have focused on publicly available related activity datasets for this proof-of-the-concept (PoC) study. We have used anonymous public PMData sports logging dataset for 16 participants, available at "Simula Website" [26]. activity data for aged, children, bodybuilders, and pregnant women is beyond this study's scope. The activity dataset has been collected with Fitbit Versa 2 fitness smartwatch in the PMSys sports logging mobile application [26]. The dataset reveals multiple features; however, we have selected the step feature for univariate time-series prediction. "Steps.json" shows number of steps per minute. Therefore, we have converted it to a daily step count for daily step forecasting.

B. Data Processing

Time-series data is strictly sequential; however, highly prone to stationarity, auto-correlation, trend, and seasonality. We have used the Augmented Dicky-Fuller (ADF) hypothesis test [23] with autolog = 'AIC' and regression = 'CT'/'C' to verify the stationarity of data. We have used the seasonal decomposition with model = 'additive' or 'multiplicative' to analyze the data's trend, seasonality, and residual components. We have changed the non-stationary data to stationary with the difference transform method. Moreover, we have used an autocorrelation 2D plot to observe the lag value vs. correlation between -1 and 1, and partial autocorrelation 2D plot with limited lag value. We also used the mean imputation method to handle missing data in time-series.

C. Deep Learning Models for Step Forecasting

Various forecasting models are available for the time-series forecasting; however, we can't determine "a priori" which one will work the best. It requires huge data for training, validation, and testing. We have used three univariate LSTM forecasting models - Vanilla, Bidirectional, and Stacked. The models are available in the Keras package. LSTMs are a particular category of Recurrent Neural networks (RNN) [27] and are used to overcome exploding gradients and vanishing gradients problems of the RNN. The states of a Vanilla LSTM model are summarized in Textbox 3. The forget gate identifies and removes redundant information from the cell state with a sigmoid layer. The input gate adds new information to the cell state. The memory cell candidate gate adds the output of forgetting and input gate to achieve a new cell state. Univariate time-series forecasting problems consist of a single series of observations. A model must learn from past observations to predict the next value in the sequence. The Multi-step time-series forecasting problem requires a prediction of multiple time steps into the future. The Bidirectional LSTM allows learning the input sequence forward and backward and concatenating both interpretations. The Stacked LSTM model consists of multiple hidden layers of LSTM units on top of one another and a predictive output layer.

TEXTBOX 3. STATES OF A VANILLA LSTM MODEL WITH A FORGET GATE.

Step 1: Forget gate: F(t) = σ(g) (W(f)*x(t) + U(f)*h(t-1) + b(f))
Step 2: Input gate: I(t) = σ(g) (W(i)*x(t) + U(i)*h(t-1) + b(i))
Step 3: Output gate: O(t) = σ(g) (W(o)*x(t) + U(o)*h(t-1) + b(o))
Step 4: Memory cell candidate: C'(t) = σ(c) (W(c)*x(t) + U(c)*h(t-1) + b(c))
Step 5: Memory cell: C(t) = F(t)*C(t-1) + I(t)*C'(t)
Step 6: Shadow state: H(t) = O(t)*tanh(C(t))
Step 7: Cell output: Y(t) = H(t)
Initially, C (0) = H (0) = 0
W: The weights of the input and recurrent connections
C(t) ∈ R^h is not a single unit of one LSTM cell; however, contains h LSTM cell's units.
σ(g): Sigmoid function, σ(c): Hyperbolic tangent function (tanh)

D. Model Training, and Testing

We have scaled our step data with the MinMaxScaler (mean = 0 and standard deviation = 1) with a feature range ∈ {0,1}. We have calculated a timestep value as a difference between the training set's length and the training data's size. The timestep value is acted as n_steps with n_features = 1. The input shape to the initial LSTM layer consisted of n_steps and n_features. We have used the mean-squared-error (MSE) loss function in the LSTM model compilation and did one-hot encoding on the prediction class variable. We have used the "ADAM" optimizer due to its space and time efficiency. We have used validation split as 0.5 with verbose = 0 and "ReduceLROnPlateau" callback to abridge the learning rate (α), thereby improving the model performance. The steps have been used for LSTM-based forecasting are in Textbox 4. Each model has consisted of 100 neurons, and we have evaluated each model against 200 epochs with a batch size of 50. We have executed each model for five times to capture mean performance score for comparison. We have used loss history for each model against each dataset to compare training and testing loss over 200 epochs.

TEXTBOX 4. STEPS FOR LSTM-BASED FORECASTING.

Step 1: Load datasets and data pre-processing (feature scaling)
Step 2: Define number of time steps, n_steps
Step 3: Split univariate time-series
Step 4: Split data for training and testing with a ratio of 80:20
Step 5: Create LSTM model with activation function
Step 6: Compile the model with optimizer and loss function
Step 7: Fit the model with training data
Step 8: Predict next day step count based on individual activity dataset
Step 9: Update Step 8 with a growing list to capture next 7-days forecast
Step 10: Store the trained model for further re-training and forecast

D. Evaluation Metrics

In this study, the performance of each forecasting model has been evaluated against RMSE [2][22][28] as it gives greater prominence to the highest errors and does not result in bias. Furthermore, we have used other metrics such as forecast bias (FB), RSD, and model execution time. The mean forecast error is known as FB. If forecast error = 0, the forecast has no error or perfect skill for that prediction. Over prediction if prediction variance < 0, model is unbiased if prediction variance = 0 or ≈ 0 [22]. FB can be positive or negative. RSD is used to statistically describe the difference in the standard deviation of observed values versus standard deviations of estimated values.

Furthermore, our ontology model has been evaluated against reasoning time, and SPARQL query execution time in Protégé and Jena Fuseki server.

IV. EXPERIMENTAL RESULTS AND DISCUSSION

This section describes - *first*, an experimental set-up to run this experiment, *second*, a comparative performance analysis of LSTM forecasting models on PMData datasets (see Table III); *third*, the selection of the best performing forecasting model and step prediction using that model; *forth*, the determination of activity type and the assignment of corresponding activity score; *fifth*, the generation of tailored activity recommendation messages for goal achievement (see Table IV for "P-1").

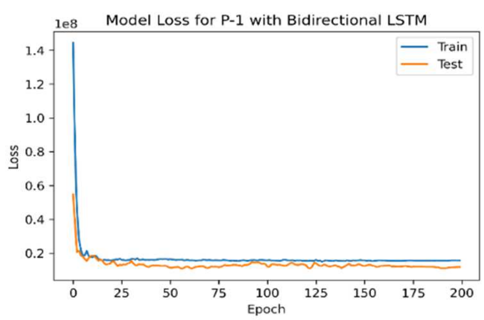
We have used Python 3.8.5 supported language libraries, such as Pandas (v. 1.1.3), NumPy (v. 1.21.2), SciPy (v. 1.5.2), Matplotlib (v. 3.3.2), Seaborn (v. 0.11.0), Plotly (v. 5.2.1), and

Keras (v. 2.6.0). We set up the Python environment in the Windows 10 operating system using Anaconda distribution and used the Jupyter Notebook v. 6.4.5 with 16GB RAM, 64-bit architecture with 512 GB storage for the model compilation, training, testing, and data visualization. Due to the low data volume, we performed the entire experiment on the central processing unit (CPU). We have used Protégé 5.x open-source editor for ontology design and implementation. Moreover, we have used the Jena Framework to query Ontology class, predicate, subject, and object and captured the corresponding execution time. We have used Hermit reasoner (v. 1.4.3.x) for Ontology reasoning and consistency checking.

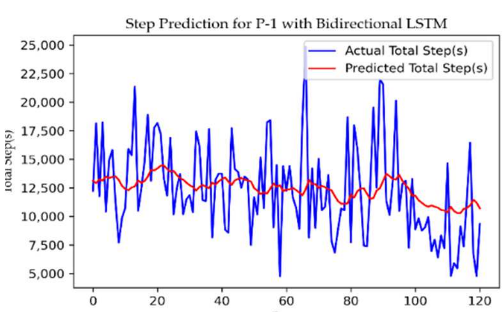
Table III shows that the Vanilla model is the fastest and the Stacked model is the slowest for the adopted PMData datasets based on the model execution time. After removing non-stationarity in PMData, the Bidirectional model has produced the best prediction accuracy against the RMSE and RSD scores. The Vanilla model has made the best |FB| score. Therefore, comparing the overall performances, we used the Bidirectional model for the next 7-days step prediction (see Table IV). For example, the structure of the Bidirectional LSTM model with layers and parameters, its performance, and prediction results for the "P-1" dataset are depicted in Figure 3. The use of the Bidirectional model reduced mean RMSE by 11.4% and 11.6%, and mean RSD by 14.0% and 14.2% compared to the Vanilla and Stacked models.

TABLE III
MEAN PERFORMANCE ACCURACY OF THE LSTM MODELS FOR ALL PARTICIPANT (P) DATA

Mean	Vanilla	Stacked	Bidirectional
RMSE	4537.3	4541.7	4069.7
| FB |	234.0	244.0	240.0
RSD	4574.7	4580.4	4011.0
ET	149.2 sec.	232.6 sec.	211.8 sec.



(a)



(b)

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
bidirectional (Bidirectional (None, 200)               81600
_________________________________________________________________
dense (Dense)                (None, 1)                 201
=================================================================
Total params: 81,801
Trainable params: 81,801
Non-trainable params: 0
```

(c)

Fig. 3. (a) The model loss, (b) the step prediction for "P-1" dataset, and (c) the parameters in the Bidirectional LSTM.

To show the practical usefulness of the forecasting model, we have conceptualized the design of an eCoach prototype system to generate customized recommendation messages to achieve personal activity goals, using the best performing Bidirectional LSTM forecast model with an Ontology model. In Table IV, we have used activity data "P-1" over 152 days for forecasting as an example. We divided 152 days of "P-1" into two segments – a. 145 days of data for training, and b. last seven days of data for forecasting. The prediction analysis helped the eCoach system to generate a tailored recommendation on a theoretical basis (see Table IV) based on the SPARQL query execution on top of the semantic rules *((Rules on activity variables) IMPLIES (Proposition variable))* in the rule base. The predicted activity score is close to the actual activity score (see Table IV). Therefore, a Bidirectional LSTM model-based forecasting can be effective in activity eCoaching. Here, we have shown the process of tailored recommendation generation for a single participant, though it can be repeated for the remaining 15 participants.

TABLE IV
DAILY STEP PREDICTION FOR A WEEK FOR "P-1", CORRESPONDING ACTIVITY TYPE, ACTIVITY SCORE, AND RECOMMENDATION MESSAGES

P	Days	Best Model	Predicted (P) vs. Actual (A) steps	Predicted vs actual activity type
P-1	Day-1	Bidirectional LSTM RMSE = 3900.0 |FE| = 102.3 RSD = 3933.0 ET = 247.6 sec.	P = 9241, A = 9121	A (2) vs A (2)
	Day-2		P = 9085, A = 7357	A (2) vs LA (1)
	Day-3		P = 10735, A = 11532	MA (3) vs MA (3)
	Day-4		P = 9783, A = 16448	A (2) vs HA (4)
	Day-5		P = 7853, A = 6698	A (2) vs LA (1)
	Day-6		P = 7135, A = 4761	LA (1) vs S (0)
	Day-7		P = 7741, A = 9351	A (2) vs A (2)
predicted		$\sum$ P-score = (1+2+1+3+2+1+1) = 14 (*Goal: 21*)		
actual		$\sum$ A-score = (2+1+3+4+1+0+2) = 13 (*Goal: 21*)		
eCoach text Recommendation on 146[th] day for the next 7 days based on P-score		***Informal message:*** *"Based on your current progress you will be 7 points behind to reach your weekly goal. Try to improve your activity level to improve your daily and weekly progress."* ***Formal message:*** *"The weather forecast for this week is Sunny. Therefore, you can perform more outdoor activities."*		

We have used OWL full to read ontology in Jena in the "TTL" format and estimated the reading time to 1.0-1.2 seconds. Moreover, we used "In-memory" storage, "optimized rule-based reasoner OWL rules" and the Jena framework to query the ontology class, ontology, predicate, subject, and object of each sentence in <1.0 seconds, <1.5 seconds, and <1.5 seconds, respectively. The estimated Ontology reasoning time has been ≈ 1.0 seconds.

The individual data collection, step prediction, and tailored recommendation generation are continuous according to the eCoach feedback cycle (see Figure 1). In authentic coaching, to accomplish a weekly goal, the eCoach module will generate personalized recommendations based on the activity outcome on each day (however, it depends on personal preferences) and followed by a predictive analysis to motivate participants to converge on their weekly or monthly activity goals. However, it is a conceptual thinking. In our future study, we will validate the effectiveness of the concept with actual participants.

V. Conclusion

This study has shown a direction to use prediction technology, Ontology, and rule base to design an eCoach system to generate meaningful, personalized activity recommendations to manage activity goals. For improving physical activity with wearable activity sensors and digital activity trackers, eCoach features can be encouraging. The concept of univariate time-series prediction exists; its application with a rule-base for activity eCoaching is novel. We will enhance our activity eCoaching with statistical methods and more activity features for meaningful hybrid activity recommendation generation.

References

[1] Physical inactivity a leading cause of disease and disability, warns WHO. Webpage: https://www.who.int/news/item/04-04-2002-physical-inactivity-a-leading-cause-of-disease-and-disability-warns-who.

[2] A. Chatterjee, et al, "Identification of risk factors associated with obesity and overweight—A machine learning overview," Sensors, 2020, vol. 20(9), pp. 2734.

[3] C. Tudor-Locke, et al, "How many days of pedometer monitoring predict weekly physical activity in adults," Prev. Med. (Baltim.), 2005, vol. 40, pp. 293–298.

[4] C.E. Matthews, et al, "I.S. Sources of Variance in Daily Physical Activity Levels in the Seasonal Variation of Blood Cholesterol Study," Am. J. Epidemiol, 2001, vol 153, pp. 987-995.

[5] B. Gardner, et al, "How to reduce sitting time? A review of behaviour change strategies used in sedentary behaviour reduction interventions among adults," Health Psychol. Rev., 2016, vol. 10, pp. 89–112.

[6] P.R.A Baker, et al, "C. Community wide interventions for increasing physical activity," Sao Paulo Med. J., 2011, vol. 129, pp. 436–437.

[7] D.E. Conroy, et al, "Lifestyle intervention effects on the frequency and duration of daily moderate-vigorous physical activity and leisure screen time," Heal. Psychol., 2017, vol. 36, pp. 299–308.

[8] L.W.C. Ng, et al, "Does exercise training change physical activity in people with COPD? A systematic review and meta-analysis," Chron. Respir. Dis., 2012, vol. 9, pp. 17–26.

[9] T. B. Dijkhuis, et al, "Personalized physical activity coaching: a machine learning approach," Sensors, 2018, vol 18(2), pp. 623.

[10] S. Schoeppe, et al, "Efficacy of interventions that use apps to improve diet, physical activity and sedentary behaviour: A systematic review," Int. J. Behav. Nutr. Phys., 2016, vol. 13, pp. 127.

[11] C.R.L Beishuizen, et al, "Web-Based Interventions Targeting Cardiovascular Risk Factors in Middle-Aged and Older People: A Systematic ReviewandMeta-Analysis," J.Med. Internet Res., 2016, vol. 18, e55.

[12] A. Chatterjee, et al, "Digital Interventions on Healthy Lifestyle Management: Systematic Review," Journal of Medical Internet Research, 2021, doi: 10.2196/26931.

[13] K. Mercer, et al, "Behavior Change Techniques Present in Wearable Activity Trackers: A Critical Analysis," JMIR mHealth uHealth, 2016, vol. 4, e40.

[14] M. Duncan, et al, "Activity Trackers Implement Different Behavior Change Techniques for Activity, Sleep, and Sedentary Behaviors," Interact. J. Med. Res., 2017, vol. 6, e13.

[15] S. Qiu, et al, "Step Counter Use and Sedentary Time in Adults: A Meta-Analysis," Medicine (Baltimore), 2015, vol. 94, e1412.

[16] A. Stephenson, et al, "Using computer, mobile and wearable technology enhanced interventions to reduce sedentary behaviour: A systematic review and meta-analysis," Int. J. Behav. Nutr. Phys. Act., 2017, vol. 14, pp. 105.

[17] A. Chatterjee, et al, "Human Coaching Methodologies for Automatic Electronic Coaching (eCoaching) as Behavioral Interventions With Information and Communication Technology: Systematic Review," Journal of Medical Internet Research, 2021, vol. 23(3), e23533.

[18] H. Rutjes, et al, "Understanding Effective Coaching on Healthy Lifestyle by Combining Theory-and Data-driven Approaches," PPT@ PERSUASIVE, 2016, pp. 26-29.

[19] J. Wang, et al, "Recognizing human daily activities from accelerometer signal," Procedia Eng., 2011, vol. 15, pp. 1780–1786.

[20] X. Li, et al, "Digital Health: Tracking Physiomes and Activity Using Wearable Biosensors Reveals Useful Health-Related Information," PLoS Biol., 2017, vol. 15, e2001402.

[21] C. Catal, et al, "On the use of ensemble of classifiers for accelerometer-based activity recognition," Appl. Soft Comput. J., 2015, vol. 37, pp. 1018–1022.

[22] Z.S. Maman, et al, "A data-driven approach to modeling physical fatigue in the workplace using wearable sensors," Appl. Ergon., 2017, vol. 65, pp. 515–529.

[23] A. Chatterjee, et al, "Statistical explorations and univariate timeseries analysis on COVID-19 datasets to understand the trend of disease spreading and death," Sensors. 2020 Jan;20(11):3089.

[24] A. Chatterjee, et al, "An automatic ontology-based approach to support logical representation of observable and measurable data for healthy lifestyle management: Proof-of-concept study," Journal of Medical Internet Research. 2021 Apr 9;23(4):e24656.

[25] A. Chatterjee, A. Prinz, "OntoRecoModel: Ontological Modeling of Personalized Recommendations for Physical Activity Coaching," JMIR Medical Informatics. 21/04/2022:33847 (forthcoming/in press).

[26] V. Thambawita et al, "Pmdata: a sports logging dataset," 11th ACM Multimedia Systems Conference, 2020, pp. 231-236.

[27] P. Khatiwada et al, "Automated human activity recognition by colliding bodies optimization-based optimal feature selection with recurrent neural network," preprint arXiv:2010.03324. 2020 Oct 7.

[28] A. Chatterjee et al, "Comparing Performance of Ensemble-Based Machine Learning Algorithms to Identify Potential Obesity Risk Factors from Public Health Datasets," In Emerging Technologies in Data Mining and Information Security 2021 (pp. 253-269). Springer, Singapore.