Turkish Online Journal of Qualitative Inquiry (TOJQI)
Volume 12, Issue 7, July 2021: 10184 - 10194

Research Article

Artificial Neural Networks based Smart Routine Plan

Dr. P. Sanjeeva Rayudu[1], Dr.Kalahasthi Neelima[2], Dr. Sampada Gulavani[3], Deepak Kori[4], J V S Arundathi[5], K G Kharade[6]

[1] Professor of ME, CREC (Autonomous), Tirupati, AP- India, dr.sanjeev168@gmail.com

[2]Professor and Head of EEE, Vignana Bharathi Institute of Technology, India, neelimarakesh@gmail.com

[3]Associate Professor, Dept. of MCA, Bharati Vidyapeeth (Deemed to be University) Institute of Management , Kolhapur, India, sampada.gulavani@bharatividyapeeth.edu

[4]Assistant Professor, Dept. of IT, IET, Dr. Rammanohar Lohia Avadh University, Ayodhya, India, korideepak@gmail.com

[5]Research Scholar, Dept. of CSE, K L Deemed to be University, Vaddeswaram, India, jvs.arundathi@gmail.com

[6]Assistant Professor, Dept. of Computer Science, Shivaji University, Kolhapur, Maharashtra, India, kgk_csd@unishivaji.ac.in

Abstract— Our daily routines have been drastically altered as a result of the breakout of COVID-19, making it difficult for most of us to cope with boredom, stress, and unpredictability. A smart routine planner was born, an application that helps users set realistic goals and prepare for them. By monitoring user efficiency, eating and sleeping habits, exercise routines, hobbies, and other leisure activities, we want to create a perfect balance between work and pleasure.

Keywords—Personalized routine planner, task scheduler, data analysis, machine learning, artificial neural networks.

I. Introduction

There are several applications on the web categorized as task managers, time-table schedulers, performance boosters, and so on. But manually creating & managing one's routine on a daily basis using several applications becomes very tiring & time-consuming. And most of these applications usually focus on solving just one or a few problems only. They usually allow users to add, update, delete tasks, set reminders, evaluate users' efficiency, and provide performance statistics. Unlike our application which not only analyzes users' routine but also ensures that they follow a healthy one.

Smart Routine Planner will allow users to either create plans manually or choose auto-generated plans. The planner will then arrange necessary tasks in a logical order and suggest a daily plan of action. Another important feature is that in case the user is unable to complete a task on time then it suggests rescheduling it to a free slot later that day.

Moreover, it will run a daily analysis on the user's routine, look for areas of improvement and make appropriate suggestions in case the user is inactive, eats in unhealthy intervals, lacks sleep, exercise, leisure, or hobbies. This is achieved through data analysis and artificial neural networks.

In this paper, we mainly focus on the following objectives:

(1) Provide users with an 'automatically generated personalized schedule' based on their lifestyle pattern, plan ratings, and history of completion of previous plans.
(2) Reschedule overdue tasks to a later time that day, if possible.
(3) Suggest an optimal sleep schedule in case the user's routine deprives him/her of sleep.
(4) Recommend appropriate hobbies as per users' interests by comparing their age, marital status, occupation, and region (optionally) with other users of the same age group.
(5) Encourage users to include physical activities and refreshment breaks in their daily routine.
(6) Real-time performance statistics, to help users visualize their daily, weekly & monthly progress.
(7) Take feedback from users in order to improve the application's scheduling performance.

II. Literature Survey

[1] In this paper, the authors have developed an application to help users achieve their daily physical activity target by analyzing their previous activity data. Daily goals are set by specifying the number of steps the user desires to walk that day. The user's success probability of achieving the target is calculated on an hourly basis. This computation is done using pre-trained models using Neural Networks, Support Vector Machine, and Logistic Regression algorithms. But this application only focuses on the user's fitness.

[2] In this paper, the authors propose the need for a tool that can analyze and manage the distribution of time towards meetings held by corporations, to ensure regularity and maximize productivity. So, they came up with MineTime Insight, a visual analytics tool that reveals hidden patterns and meeting distributions. But this application is suitable only for working professions.

[3] In this paper, the authors discuss a scheduling system called Intelligent Daily Scheduler, a mobile application that schedules personalized weekly timetables for users. The application collects data about users' lifestyles and uses the Deep Neural Network (DNN) model to predict patterns in their daily schedule. They have developed a scheduling algorithm that minimizes task juggling and allocates continuous long intervals of time rather than small intervals. But it does not ensure whether users follow healthy routines or not.

[4] In this paper, the authors have developed a system that assists users in planning their daily or weekly schedules. The phi correlation coefficient is calculated for each task type by performing a correlation analysis between task type and user rating. Then ranks are assigned to each task type and the two highest positively ranked task types are recommended.

[5] Intelligent Timetabling Scheduler is a web-based application that generates exam timetables and chooses an arena considering the number of students attempting the exam, it also appoints a supervising lecturer. For semester timetabling, it ensures that no professor or student is assigned to more than one class at the same time. This application is only suitable for academic institutions.

[6] In this paper, the authors propose an algorithm to assist users in scheduling their daily routines according to certain constraints and dependencies. Temporal constraints ensure that no two activities start at the same time or intersect during their execution time. They have also used

backtracking and back-jumping to ensure that all constraints are satisfied.

[7] In this paper, the authors propose an algorithm that finds an optimal agenda to fix appointments and finish tasks. It clusters the existing list of tasks, bins them according to spatial-temporal proximity to the new task, finds an optimal solution for the subset of tasks within the cluster closest to the new task, and expands the search area until a valid schedule is reached.

[8] In this paper, the authors have developed an Intelligent Calendar System that handles temporal inconsistencies in scheduling events. They have used distance-based heuristics for solving temporal inconsistencies between events based on their attributes.

III. System Overview

We have created Smart Routine Planner as a mobile application so that users will be able to easily use it on their phones. We have used artificial neural networks to perform automatic plan generation and suggest hobbies.

To automatically generate the user's routine we have implemented a two-phase training approach. Firstly, we train a base model to predict the types of tasks to be done for the given day. Then we train a secondary model to predict the actual tasks, given the type of tasks to be performed.

For hobby suggestions, we train a single model to suggest hobbies based on users' age, marital status, occupation, and optionally region. The top three hobbies with the highest probability are recommended to the user.

As discussed in [9], pattern classification becomes more complex as data dimensionality increases. But with deep learning algorithms like Deep Artificial Neural Networks, it becomes easier to make predictions and obtain satisfyingly accurate results by training models on large datasets.

IV. Methodology

A) Routine Planner

Firstly let's discuss the primary objective of our application, which is to help users create feasible plans, analyze and monitor their progress. This section contains the following subsections: 1) Data collection, 2) Create Plans, 3) Monitor progress, and 4) Reschedule pending tasks.

1) Data Collection

It is mandatory for new users to manually create plans for the first six days. This data will serve as an initial dataset for automatic plan prediction in the future.

2) Create Plans

i) Manually Create Plan

Manually creating plans contains the following four steps: a) Obtain plan data b) Determine idle time c) Analyze plan & derive insights d) Provide suggestions

a) Obtain Plan Data

Users should mention the start-time and end-time of each task and also select a category that represents the task type. The categories are as follows: sleep, fitness, refreshment, work, chores, hobby, leisure, social, others, and idle. Not more than one task can be assigned at a given time.

Task priority can be set to either low or high and the default priority for all tasks is low. All tasks are marked incomplete by default and can be changed to completed if finished.

b) Determine Idle Time

An entire day (i.e. 24 hours) is divided into 5-minute intervals thus creating 288 time slots numbered from 0 to 287. By default, all 288 time slots are marked unassigned. The plan data obtained in the previous phase is now used to assign tasks to their respective time slots.

Given the start-time and end-time in 24 hour (hh:mm) format, we can determine the start and end time slots using the formula given below:

$$Slot\ No = (Hour * 12) + (Minute / 5),$$

After this process is completed, we know that the unassigned time slots are analogous to the user's idle time and are assigned the category 'idle'.

c) Analyze Plan & Derive Insights

Now we have information about the duration of each task in terms of the number of time slots. Since each time slot corresponds to 5 minutes we can determine the duration in both hours and minutes using the given formula:

$$Task\ duration\ (minutes) = No.\ of\ time\ slots * 5$$

This information is used to determine the time spent on recreational activities, refreshment breaks, fitness, and rest.

Then the user's idle-time intervals are also included to be a part of the existing plan. For this, we use the following formulas to determine the start-time and end-time of each idle-time interval in 24 hour (hh:mm) format.

Let 's' represent time slot number,
hour = integer value of (s / 12)
*minute = (s – (hour * 12)) * 5*

d) Provide Suggestions

Let's assume the user's total sleep time is 'x' then if 'x' is less than 7 hours she/he is suggested to sleep for another (7 – x) hours in their free time. Similarly, if the user does not include fitness in their routine, then the application advises them to set aside at least 30 minutes for the same. In both cases, idle time intervals with a duration of 30 minutes or more are also included with the suggestion.

Shown below are the optimum time intervals in which the user is expected to take refreshment breaks. If they fail to do so, then they are suggested to include a refreshment break for that particular time of the day.

Time of the Day	Time in hh:mm	Time Slots
Morning	*(00:00 – 12:00)*	*0 - 144*
Mid-day	*(11:00 – 18:00)*	*132 - 216*
Evening	*(17:00 – 24:00)*	*204 - 287*

Table 1. Time intervals for morning, mid-day, and evening

ii) Automatically-Generated Plan

There are four phases in automatic plan generation [10], [11], they are as follows: a) Preparing the dataset b) Training the base model to predict task categories c) Training the secondary model to predict actual tasks d) Validating predicted plan

a) Preparing the Dataset

Those plans with completion scores >= 50 % and ratings >= 3 are selected to be included in the dataset.

Selected plans are then converted into a time slot format. Since each day is divided into 5-minute intervals of 288 time slots thus the minimum size of the dataset is 1728 corresponding to six days' plans.

The dataset includes the following fields: day, time slot, category, and task. If the user has fewer than 14 days plans, then the value of day will be replaced with the current day, for the entire dataset.

b) Training the Base Model to Predict Task Categories

In order to train the model to predict the task category given the day, we have to consider only the day, time slot, and category from our dataset. Day and time slot serve as the input features to our model and the category or type of task is the target feature.

We have used Artificial Neural Networks to build the model. The input layer contains 20 nodes and the activation function used is mish, a self regularized non-monotonic neural activation function that out-performs both swish and ReLU in terms of accuracy.

There are three hidden layers containing 20, 15, and 15 nodes respectively. Mish activation function is used for the first and second hidden layer and the last hidden layer uses softplus, a smooth approximation to the ReLU activation function, that constrains the output of a machine to always be (+ve).

Lastly, the output layer contains 10 nodes corresponding to the number of task categories. The activation function used is softmax, which determines the probabilities of each category. That with the highest probability is the target category.

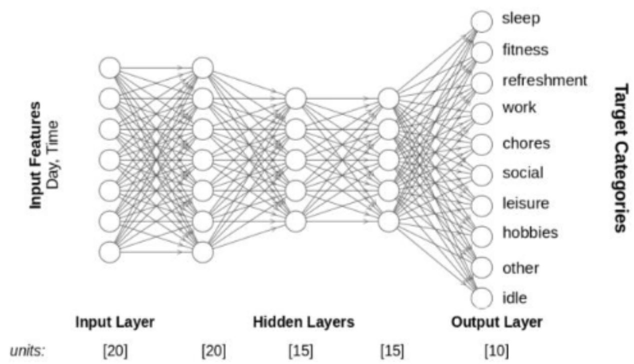

Figure 1. Base model for task category prediction

Loss is calculated using categorical-cross-entropy and the optimizer used is adam with 0.01 as the learning rate. Accuracy by default is moderate (i.e. 78%), but can also be set to high (i.e. 85.55%). The model trains until the required accuracy is reached. The maximum no. of epochs is 200 with a batch size of 10. If the required accuracy is not met but the accuracy attained >= (required accuracy – 15%) then the model is considered as final. Else the required accuracy is reduced by 5% and the model is rebuilt, this will be the final model irrespective of the accuracy attained.

The model is then used to predict the category of the task to be performed for every time slot of the current day. This data is then used by the secondary model to predict the actual task to be performed for every time slot of the current day given the category or type of task.

c) Training the Secondary Model to Predict Actual Tasks

This model is similar to the base model in terms of the number of layers and their activation functions, but the number of nodes in each of the layers is variable. The input layer uses mish activation function and consists of 2n nodes, the first two hidden layers use mish activation function and the last hidden layer uses softplus activation function. They contain 2n, 1.5n, 1.5n nodes respectively. The output layer uses softmax activation function and contains n nodes where n is the number of distinct tasks.

In order to train the model to predict the actual tasks given the day, time slot, and type of task (i.e. category) we have to consider day, time slot, category, and task from our dataset. Day, time slot, and category serve as the input features to our model and task is the target feature.

Loss is calculated using categorical-cross-entropy and the optimizer used is adam with 0.01 as the learning rate. The required accuracy is initially set to 93%. The model trains until the required accuracy is reached. The maximum no. of epochs is 50 with a batch size of 5. If the required accuracy is not met but the accuracy attained >= (required accuracy – 25%) then the model is considered as final. Else the required accuracy is reduced by 3% and the model is rebuilt, this will be the final model irrespective of the accuracy attained.

The model is then used to predict the actual tasks for every time slot of the current day, given the category of tasks predicted by the base model.

d) Validating Predicted Plan

The predicted plan is then analyzed to ensure whether it includes optimum sleep, regular refreshment breaks, and leisure time.

If sleep is less than 4 hours ~ 48 time slots (i.e. 48 * 5 minutes) then the plan is searched from the beginning for 6 consequently idle time slots (i.e. 6 * 5 minutes of idle time) and if found these idle time slots are allotted to sleep. This process continues until either the condition is satisfied or there are no more idle time slots.

Similarly, if the plan has not allotted refreshment breaks in the morning, mid-day or evening, then the first occurring idle time slot in that particular range is allotted to refreshment. The second time slot is also allotted only if it is idle. Refer Table 1. Time intervals for morning, mid-day, and evening.

Lastly, if the plan has not included any leisure, hobby, or social activity, then the plan is searched for 4 consequently occurring idle time slots (i.e. 4 * 5 minutes), if found this idle time is dedicated to leisure.

The validated plan is then presented to the user after converting the duration in time slots to actual start and end time in 24 hour format (hh:mm).

3) Monitor Progress

Progress scores are calculated as and when tasks are marked as completed. Weights are assigned to tasks based on their priority. The weight of tasks with low priority is 1 and that of those with high priority is 2.

$$Total\ weight = (no.\ of\ tasks\ with\ high\ priority)*2 + (no.\ of\ tasks\ with\ low\ priority)$$

The progress score is initially zero. Whenever a task is marked as completed, the progress score is incremented by the weight of that task. The completion percentage is then calculated using the given formula.

*Completion % = (Cumulative progress score / Total weight) * 100*

4) Reschedule Pending Tasks

Rescheduling of tasks overdue can be performed only on the current day's plan. Rescheduling is done based on task priority, those with higher priority are rescheduled before those with lower priority.

An overdue task is rescheduled to a later time that day if an idle time interval of duration >= (task duration – 15 minutes) is available. Rescheduling of tasks may or may not be possible as it depends on the number of overdue tasks, their duration and the amount of idle time currently available.

B) Hobby Finder

This feature suggests users new hobbies to pursue. This section includes the following subsections: 1) Data collection, 2) Preparing the dataset, and 3) Training the hobby predictor model.

1) Data Collection

We collected information about age, marital status, occupation, region, and hobbies from around 200 people between the ages of 14 to 65. And on registration, we obtain users' age, marital status, and other details if specified. The default value for occupation is 'others'. Region and hobbies are null if not mentioned.

2) Preparing the Dataset

Let's consider recommending hobbies for user X. So for the dataset, we will consider only those with ages ranging from (X's age – 5) to (X's age + 5) from the entire data collected via survey.

3) Training the Hobby Predictor Model

The input features to the ANN model are age, marital status, occupation, and optionally region. And the target feature is hobby.

The model consists of an input layer that uses mish activation function and consists of 48 nodes. There are three hidden layers, the first two hidden layers use mish activation function and the last hidden layer uses softplus activation function. They contain 48, 36, 36 nodes respectively. The output layer uses softmax activation function and contains 24 nodes that correspond to 24 types of hobbies.

Loss is calculated using categorical-cross-entropy and the optimizer used is adam with 0.01 as the learning rate. The required accuracy is initially set to 58%. The model trains until either the required accuracy is reached or the loss < 1. It also stops if the same accuracy is consequently obtained 12 times. The maximum no. of epochs is 500 with a batch size of 10.

The trained model is then used to predict hobbies given the user's age, marital status, occupation, and optionally region. From the list of hobbies suggested we remove those hobbies that are already in the user's current hobby list. And suggest a maximum of three hobbies (i.e. those with the highest probabilities).

V. DESIGN

On registration, users are asked to compulsorily mention their age and marital status and other details like hobbies, region, and occupation, which are optional. This information is used to recommend hobbies.

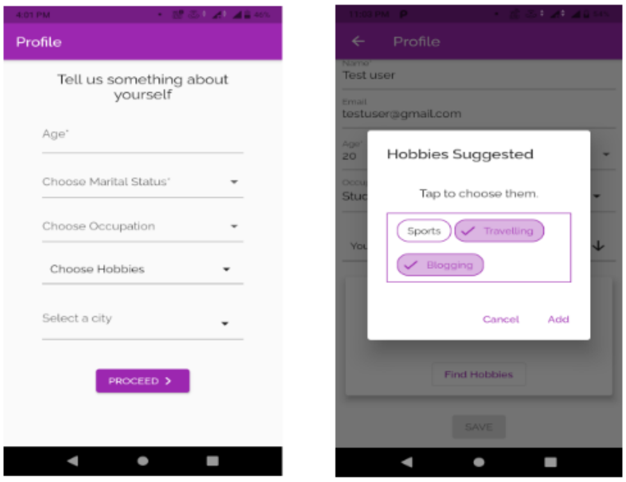

Figure 2.Registration(left) & hobbies recommended(right)

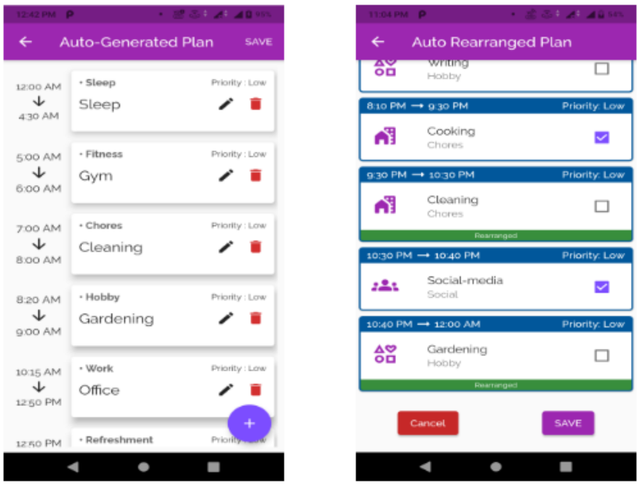

Figure 3. Automatically generated plan(left) and automatically rescheduled plan(right)

Completion scores are used to visualize users' performance. Performance statistics can be viewed daily, weekly, and monthly.

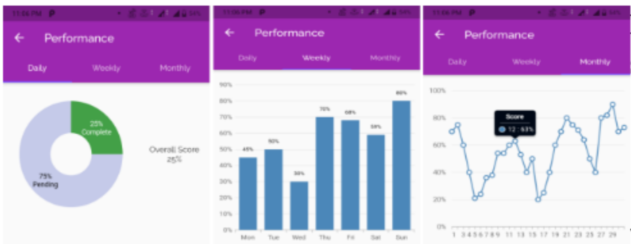

Figure 4. Daily, weekly & monthly performance statistics.

VI. RESULT

The application performs exceptionally well when it comes to automatically generating daily routines. The results are as follows:

i) Base model for task category prediction:

Quality	Accuracy	Loss	No. of Epochs

Moderate	78.0%	0.57	69
High	79.0%	0.51	126

ii) Secondary model for actual task prediction:

Quality	Accuracy	Loss	No. of Epochs
Moderate	93.0%	0.59	34
High	95.18%	0.21	8

The recommend hobbies model works fairly well considering the fact that hobbies vary greatly from person to person and only partially depends on factors like age, marital status, occupation, region, and so on.

iii) Recommend hobbies model:

Include Region	Accuracy	Loss	No. of Epochs
Yes	54.1%	1.13	499
No	60.0%	1.05	402

VII. Conclusion

Proper planning helps in reducing stress to a great extent. A feasible plan with attainable goals helps ease one’s mind by giving them confidence and direction.

Thus, the Smart Routine Planner is like a supportive friend who helps one to achieve their goals by making the best use of their time, keeping work and leisure in perfect balance, and also ensures that they follow a healthy routine. Thereby, implementing a structure to their day that not only gives them a sense of control but also improves their focus, organization, and productivity.

References

[1] Z. Li, S. Das, J. Codella, T. Hao, K. Lin, C. Madurtime slotChen, "An Adaptive, Data-Driven Personalized Advisor for Increasing Physical Activity," in IEEE Journal of Biomedical and Health Informatics, vol. 23, no. 3, pp. 999-1010, May 2019, doi: 10.1109/JBHI.2018.2879805.

[2] M. Ancona, M. Beyeler, M. Gross and T. Guenther, "MineTime Insight: Visualizing Meeting Habits to Promote Informed Scheduling Decisions," in IEEE Transactions on Visualization and Computer Graphics, doi: 10.1109/TVCG.2019.2941208.

[3] J. Geetha, B. S. Akanksh and A. S. Koushik, "Intelligent Daily Scheduler," 2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Bangalore, 2018, pp. 2308-2312, doi: 10.1109/ICACCI.2018.8554485.

[4] M. A. Kuhail and N. S. S. Gurram, "TaskDo: A Daily Task Recommender System," 2019 International Conference on Computational Intelligence in Data Science (ICCIDS), Chennai, India, 2019, pp. 1-5, doi: 10.1109/ICCIDS.2019.8862073.

[5] T. W. Ekanayake, P. Subasinghe, S. Ragel, A. Gamage and S. Attanayaka, "Intelligent Timetable Scheduler: A Comparison of Genetic, Graph Coloring, Heuristic and Iterated Local Search Algorithms," 2019 International Conference on Advancements in Computing (ICAC), Malabe, Sri Lanka, 2019, pp. 85-90, doi: 10.1109/ICAC49085.2019.9103403.

[6] F. S. Alhaijawy and A. M. Florea, "Scheduling People's Daily Activities Using Temporal Constraints Satisfaction Problem," 2018 5th International Conference on Control, Decision and Information Technologies (CoDIT), Thessaloniki, 2018, pp. 876-881, doi: 10.1109/CoDIT.2018.8394957.

[7] Stephanie Rosenthal and Laura M. Hiatt. 2020. Human-Centered Decision Support for Agenda Scheduling. In Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS '20). International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 1161–1168.

[8] T. JiaHua and D. Zhang, "SmartCalendar: Improving Scheduling Through Overcoming Temporal Inconsistencies," 2018 IEEE 17th International Conference on Cognitive Informatics & Cognitive Computing (ICCI*CC), Berkeley, CA, 2018, pp. 30-37, doi: 10.1109/ICCI-CC.2018.8482075.

[9] I. Arel, D. C. Rose and T. P. Karnowski, "Deep Machine Learning - A New Frontier in Artificial Intelligence Research [Research Frontier]," in IEEE Computational Intelligence Magazine, vol. 5, no. 4, pp. 13-18, Nov. 2010, doi: 10.1109/MCI.2010.938364.

[10] I. Arel, D. C. Rose and T. P. Karnowski, "Deep Machine Learning - A New Frontier in Artificial Intelligence Research [Research Frontier]," in IEEE Computational Intelligence Magazine, vol. 5, no. 4, pp. 13-18, Nov. 2010, doi: 10.1109/MCI.2010.938364.

[11] Sonika M., T Babu S.B.G., Nandan D. “Systematic Observation on Non-orthogonal Multiple Access for 5th Generation Communication Technology,” ICCCE 2020. Lecture Notes in Electrical Engineering, vol 698. Springer, Singapore. https://doi.org/10.1007/978-981-15-7961-5_89.