Research of Context Awareness Based Accident Prevention During Mobile Phone Use

Bowen Sun[1], Quanlong Li[2], Yonghui Guo[3], Guocheng Li
sunbowen@hit.edu.cn[1], liquanlong@hit.edu.cn[2], guoyon9hui@gmail.com[3]

School of Computer Science and Technology, Harbin Institute of Technology, Harbin, Heilongjiang, China[123]

Abstract. In recent years, with the continuous development of technologies, people's dependence on smartphones is growing increasingly. Excessive dependence is causing a considerable number of people to face potential accidents and injuries on account of the use of mobile phones. Based on above considerations, this paper gives defines dangerous context, using context-aware technology to study the accident prevention methods in the process of using mobile phones, and the research involves three parts: human behavior awareness, spatial location awareness, and interaction mode awareness.

Keywords: dangerous context, context awareness, mobile phone use, accident prevention.

1 Introduction

With the maturity and popularity of emerging technologies such as ubiquitous computing, social networking, and the mobile Internet, a large number of new things have emerged, and smartphones are one of them. The rapid development of hardware technology has allowed smartphones to have the same capability of computing and storage as the computers of the beginning of this century. Smartphones not only perform as a communication tool but also gradually become an intelligent terminal combining sensing, computing and communication capabilities.

In this context, a wide variety of mobile applications continue to show up in people’s daily lives and work. The user’s excessive reliance on smartphones has caused the appearance of a new term, the "phubber" – the word which is used to describe a group of people who use their smartphones, tablet computers or other mobile devices with their heads down for a long time. This phenomenon is common in our daily life. For example, people nowadays often use mobile phones to browse information while having meals. What is worse, the same situation often occurs on the roads or crosswalks that are more prone to happen traffic accidents, and the appearance of phubbers has been becoming a serious threat to the personal safety for both pedestrians and drivers. According to the World Health Organization's 2015 Road Safety Report[1], since 2001, the number of deaths from road traffic accidents all over the globe has been continually on the rise. The growing phenomenon of phubbers is one of the leading causes. For this social problem, required to be solved urgently, several countries have implemented measures to address this issue, including manual monitoring, building infrastructure and completing the rules through legislation or other effective means, in order to have control on users’ use of smart devices.

The purpose of this paper is to collect data of users through the embedded sensors of smartphones. Then by combining context-aware technology with other advanced technologies (such as machine learning), to judge whether users are having the risk of accidental injury during mobile phone use. After getting the results of judgments, the phones will take the protective measures, thereby effectively controlling the occurrence of the phenomenon of phubbers as well as reducing the loss of life and property caused by it.

Based on the previous researches and applications, we define the dangerous context in the mathematical form. Then we use more intelligent methods to recognize the behavior of the mobile phone users and consider the integrated context to distinguish whether the user is in a dangerous context. What is more, with this application, the phone could make the recognition itself, which could give the alert immediately to avoid potential danger, thus to protect the users effectively.

The rest of this paper is organized as follows. Section 2 introduces the related works toward this topic. Section 3 introduces the model and algorithms we propose for the research. Section 4 gives the experimental design and results in this paper. Section 5 briefly introduces the application we implement based on the theories. Moreover, section 6 concludes the work in this paper and introduce future work as well.

MOBIMEDIA 2019, June 29-30, Weihai, People's Republic of China

DOI 10.4108/eai.29-6-2019.2282759

2 Related works

The concept of context awareness derives from the field of ubiquitous computing. Abowd et al.[2] published a paper in 1999 that gave a more general definition of context awareness, which defined it as the use of contextual information to provide relevant services or information to users (contextual information can usually be labeled using sensor data). Bangerter et al.[3] mentioned a new technique in the paper on network and device research about 5G technology, in order to provide users with higher quality services and experiences. The systems of 5G need to implement context awareness, based on the network, devices, applications, as well as the environment users locate in, using contextual information that can be perceived to provide users with personalized services. Shin et al.[4] proposed a pedestrian dead reckoning system (PDR) based on mobile phone location awareness algorithm, and the acceleration sensors in the mobile phone to estimate the user's step size and the number of steps.

Specifically, context-aware research and applications have covered multiple aspects. Josu é Iglesias et al.[5] based on the motion acceleration and positioning information collected by sensors, proposed a mobile application for activity monitoring on mobile phones, which monitors users' behavior in real time. The application evaluates users' activities throughout the day and provides them the information about their level of movements, and it considers related privacy issues.

ChangKyun Jeon et al.[6] used context-aware technology to analyze the interaction between users and mobile devices over some time and to take effective measures to prevent mobile devices from overheating by obtaining user interaction status and current mobile phone applications in use. This method can be used to solve the problem of energy consumption in a dynamic way. J. Zaldivar et al.[7] developed an application for detecting traffic accidents and ensuring driving safety by using the interface provided by the smartphone (OBD-II), by judging the current situation of the driver and the vehicle. This application gives the detection of driving safety and automatically takes emergency measures when detecting the accident. Similarly, Bankar Sanket Anil et al.[8] also proposed a project, aiming to work as an intelligent detection and warning system when the vehicular accidents happen. Fabio Martinelli et al.[9] designed and developed a context-aware mobile application consisting of a secondary system of driver situation analysis and alarm information to prevent traffic accidents. Besides, context-aware technology was widely used in encryption application design to protect user privacy[10], to prevent data loss in cloud storage services[11], and to provide intelligent control of smart life and home design[12] and other applications.

The primary research of this paper is based on context-aware technology and the prevention of accidental injury during the use of mobile phones. The main question to be answered is whether the user is in a dangerous situation caused by the use of mobile phones. If so, the intervention will be required. In this paper, we divide the solution to the whole problem into three sub-problems, namely human behavior awareness sub-problem, spatial location awareness sub-problem and interaction mode awareness sub-problem. By solving sub-problems then utilizing the output of sub-problems, we can determine whether the mobile phone holder is in a dangerous situation, and to increase the credibility for the specific situations and causes during the problem-solving. The purpose of the research is to prevent possible accidental damage caused by the use of mobile phones based on context-aware technology, thereby reducing the loss of life and property.

3 Model & Algorithm

In this part, we will give our definition and model designed for solving the problem, as well as the algorithm we implement for the three sub-problems. Specifically, this part includes the process of data collection and data preprocessing, the design and implementation of the algorithms, the integration of algorithms, and the complexity analysis.

3.1 Definition and modeling of the dangerous situation

Before the awareness and recognition of the dangerous situation, the dangerous situational awareness and recognition process should be modeled first. Taking human behavior, the interaction between the user and the mobile phone, and the spatial location of the user into consideration, we mathematically define the dangerous situation recognition process as equation (1).

$$C = (behavior，\ space，\ time)\ . \tag{1}$$

In this definition, *C* (Context) is a continuous variable indicating the output of the dangerous situation recognition model. When the value of *C* is greater than a specific threshold D_0, we consider that the user is in a dangerous situation, and the set is *D* (Danger). *behavior*, *space*, and *time* respectively represent the output of three sub-problems. *behavior* represents the atomic behavior of the phone holder, such as standing, walking, riding, driving, riding, and so on. *space* represents the current spatial location information of the phone holder, such as crossroads, crosswalks, and so on. *time* indicates the possibility that the phone holder is currently interacting with the phone. That is, the dangerous situation recognition model accepts the above three parameters as model input, and outputs the possibility that the mobile phone holder is currently in a dangerous situation. The research on the accident prevention method based on context-aware mobile phone use is the solution to the dangerous situation recognition model.

After converting the output of the three sub-models into continuous random variables, we further define the dangerous situation recognition model in the form as equation (2) and (3).

$$C = behavior * space * time \,. \tag{2}$$

$$level \ = \begin{cases} C - D_0, & D > D_0 \\ \ 0 \ \ , & D \leq D_0 \end{cases} \,. \tag{3}$$

The output of the dangerous situation recognition model is defined as the product of the output results of the three sub-models. For the level of danger, the value depends on the relationship between the output *C* of the dangerous situation recognition model and the threshold D_0. When the value of the level increases, we may increase the corresponding warning level, such as pop-up dialog reminder, phone vibration, forced lock screen, and so on.

In order to determine the threshold D_0, this paper conducts experiments under different mobile phone usage states and uses the K-Means for one-dimensional clustering to determine the parameter D_0 by using the results of the dangerous situation recognition model in various states. In order to compare the output values of the dangerous situation recognition model corresponding to the three sub-questions, the final threshold D_0 is set to 48476. Based on this threshold, the results of the dangerous situation recognition are verified by manual labeling. The accuracy of the recognition model is 96.63%, the recall rate is 87.76%, and the F1 value is 91.98%. Overall, the model has a good recognition effect on dangerous situations.

Based on the above formal definition of the dangerous situation recognition model, one of the conditions for the typical dangerous situation $d \in C$ is as follows:

$d \in D$

s.t. { *d = (behavior, space, time)* | *behavior* = {walking，riding，driving} ∩ {browsing the mobile phone，interacting with the phone}，*space* = { crossroads，crosswalks}，*time* = current_time}

In the same way, for some situations $s \in C$ defined as:

{ *d = (behavior, space, time)* | *behavior* = {taking a ride} ∩ { browsing the mobile phone，interacting with the phone }，*space* = { crossroads，crosswalks }，*time* = current_time }

Thus, *s* does not belong to a dangerous situation - $s \notin D$ or $s \in \overline{D}$. Because although the user interacts with the mobile phone and is in an accident-prone location, it is in a state of taking a ride instead of driving a car, so it will not be recognized as a dangerous situation, and other dangerous situations or non-dangerous situations are the same.

3.2 Human behavior recognition algorithm

The awareness of the atomic human behavior of mobile phone holders is an important sub-problem of dangerous situation recognition. The behavior of the phone holder and the location of the phone will reflect whether the user is using the phone or not.

Data collection. The human activity data used in this paper is collected from the Android smartphone Nexus 5X. The Android sensor coordinate system is fixed relative to the mobile phone. The location and direction of the mobile phone will affect the distribution of specific sensor data. Therefore, it is necessary to collect data from different placed states of mobile phones. For each piece of data collected, the types of data collected include the time stamp, three-axis acceleration, three-axis angular acceleration, three-axis gravitational acceleration, latitude, and longitude information, and a total of 400,000 human behavior data including 32 behavioral tags were collected.

Atomic human behavior category coding. The atomic human behavior that this article focuses on consists of human behavior, the location of the mobile phone holder and the direction in which the mobile phone holder holds the mobile phone, such as walking - with the mobile phone in hand (body side) - the mobile phone screen is facing toward the holder inside - the phone is facing down. Based on the above collection process, this paper finally determines 32 kinds of atomic human behavior categories and uses the form of one-hot encoding to encode the atomic human behavior classification. Its space overhead is equivalent to the digital encoding method, which is more space-saving than other text encoding methods. We can obtain the one-hot encoding in different group states by bit operation (and, or, shift, and so on) with extremely high efficiency. The information contains more information than the digital encoding, and the speed of obtaining the information is much higher than the text encoding. The state as described above, which is walking - with the mobile phone in hand (body side) - the mobile phone screen is facing toward the holder inside - the phone is facing down could be encoded in one-hot encoding as a sequence of 00100001000001000000001.

Data preprocessing. The original data is divided into several windows according to a certain length of time, and we perform feature extraction for each window. The predefined features include the mean of three-axis acceleration, the standard deviation of three-axis acceleration, mean of angular acceleration, the standard deviation of angular acceleration, mean of gravitational acceleration, the standard deviation of gravity acceleration and speed.

Algorithm design and implementation. The characteristics contained in a single record in the data set are all numeric types, that is, continuous random variables. Therefore, the decision tree model for human behavior perception is trained based on the CART algorithm. Two parameters, *min_split_sample* and *min_gini_gain*, are set at the same time to control the excessive growth of the decision tree and prevent the decision tree model from over-fitting the training set. *min_split_sample* represents the minimum number of samples in which the sub-nodes can continue to be divided. If the number of samples on the current node is less than this value, the node is directly used as the leaf node. After the node is divided, if the amount of decrease in the Gini index is less than *min_gini_gain*, then the node is directly used as the leaf node and no longer continues to be divided.

The random forest training could be carried out based on the above decision tree model, and the structure of random forest is controlled by setting two parameters *n_trees* (the number of decision trees in the random forest) and *n_feature_used* (the size of the feature subset used in training each decision tree). Each decision tree in a random forest can be considered as an available classifier. Assuming that the number of decision trees in a random forest is *N*, for instance (feature vector), input into a random forest, the *N* numbers of decision trees will have a classification for *N* times. Then, the classification results of *N* are generated. The random forest integrates the classification results of all decision tree classifiers and votes according to the classification results. The category with the highest number of votes will be used as the classification result of the random forest classification model. The main parameters we define for the algorithm are shown in **Table 1**, and the pseudo code of the algorithm is shown as **Algorithm 1** as follows.

Table 1. Parameters defined in Algorithm 1

Parameters	Representation
origin_dataset	Original dataset
win_len	Length of windows
n_trees	Number of decision trees in random forests
n_feature_used	The size of the feature set used to train each decision tree
feature_dataset	The dataset after feature extraction
dt_trained	Training times
rf	The collection of all decision tree root nodes in a random forest

Algorithm 1 Random forest generation algorithm

```
Input: origin_dataset, win_len, n_trees, n_feature_used
Output: rf
feature_dataset ← extract features from origin_dataset
training_set ← split feature_dataset into training and test set
feature_set ← feature set of training_set
dt_trained ← 0
rf ← []
while dt_trained < n_trees do
```

```
        sub_training_set ← bootstrap sampling for training_set
        sub_feature_set ← randomly selecting a number of n_feature_used from feature_set
        sub_training_set ← mapping sub_training_set on sub_feature_set
        rf.append(
                tree_generate(sub_training_set, 2, 0, win_len, true)
        )
        dt_trained ← dt_trained + 1
    end while
    return rf
```

Complexity analysis. The time complexity of the human behavior-aware algorithm based on the random forest is mainly composed of the time complexity of the training decision tree. For each decision tree in the random forest, the training set used by the training set contains the minority of the non-repetitive samples compared with the original training set, but the size is consistent with the original training set. The size of the feature set used is fixed to the original feature set size as *K*, and the number of decision tree classifiers in the random forest is recorded as *N*, then the time complexity of the human behavior awareness based on the random forest algorithm is $O(N*K|M||D|^2log|D|)$.

3.3 Spatial position awareness algorithm

The location of the mobile phone holder has a certain correlation with the possibility of being in a dangerous situation. The primary purpose of perceiving the spatial position is to judge the extent to which the mobile phone holder is close to the area prone to accidental injury. Based on this background the area in this paper is defined as a traffic road.

Data format and pre-processing. Information about the current spatial location of the phone holder (namely latitude and longitude, longitude, and so on) is collected by the location sensor built in the smartphone and the location service provided by the LBS service provider, and the location of the phone holder is sampled at a relatively high frequency. At the same time, by using OpenStreetMap, we can get the road network information of the area where the phone holder is located. The spatial position of the phone holder at time *t* can be defined as equation (4).

$$P = \{t, latitude, longitude, error\} . \tag{4}$$

In the definition *t* is the timestamp recorded by *P*, *latitude* is the latitude information, *longitude* is the longitude information, *error* is the precision information, represented in meter, and the smaller the *error* is, the higher the accuracy will be. When the spatial position of the mobile phone holder is periodically sampled at regular time intervals, the track of the spatial position changing can be obtained, and the spatial position changing the track of the mobile phone holder under *N* in sampling periods is defined as equation (5).

$$T = \{P_0, P_1, \dots , P_{N-1}\} . \tag{5}$$

Moreover, for arbitrary value $i \in 0,1,...,N\text{-}2$, all exist $P_{i+1} * t - P_i * t = t_0$.

Algorithm design and implementation. The initial link candidate set *initial_segments* is calculated according to the initial link candidate set selection algorithm. Next, using the current position *P* of the mobile phone holder, the position trajectory *T*, and the initial road segment candidate set *initial_segments*, a candidate road segment set with the size of *candidate_size* is generated. Moreover, the reason for using the candidate road segment set as the output is that the position is dependent on the spatial position sensing algorithm based on the location point. The distance from the point to the road segment will largely depend on the accuracy of the position point *P*. When the accuracy of the single point positioning is low, positioning errors may occur. For this consideration, among the several distances calculated by the algorithm, the smallest road segment is used as a candidate. The main parameters we define for the algorithm are shown in **Table 2**, and the trajectory-based spatial position sensing algorithm for candidate applications is shown as **Algorithm 2**.

Table 2 Parameters defined in Algorithm 2

Parameters	Representation
T	Position trajectory in the past period
initial_segments	Initial segment candidate set corresponding to *p_user*
p_user	User's current location
candidate_size	Number of candidate segments reserved
candidate_segments	The candidate segments
proximity_level	Using the results of location-based spatial location awareness to maximize the proximity of similarity between phone holders and roads segments

Algorithm 2 Spatial position awareness algorithm based on position trajectory
Input: T, nitial_segments, p_user, candidate_size
Output: proximity_level

```
1:  candidate_segments ← determine_location_by_position(
                initial_segments, p_user, candidate_size
        )
2:  max_proximity_level ← -∞
3:  for segment in candidate_segments
4:      avg_distance ← 0
5:      for P in T
6:          avg_distance ← avg_distance + distance_to_segment(segment, P)
7:      end for
8:      avg_distance ← avg_distance / len(T)
9:      k ← (segment.$P_A$.latitude - segment. $P_B$.latitude)
            / (segment. $P_A$.longitude - segment. $P_B$.longitude)
10:     $V_S$ ← (cos(arctan(k)), sin(arctan(k)))
11:     $V_T$ ← least_square_fitting(T) or mean_gradient_fitting(T)
12:     direction_matching ← $V_S$ * $V_T$
13:     proximity_level = direction_matching / avg_distance
                + 1 / segment.level
14:     if proximity_level > max_proximity
15:         max_proximity_level ← proximity_level
16:     end if
17: end for
18: return max_proximity_level
```

Complexity analysis. The time complexity of the *determine_location_by_trajectory* algorithm is *O(K+L*|T|)*, where *K* is the size of the initial segment candidate set *initial_segments*, and *L* is the size of the set *candidate_segments*. The *proximity_level* is defined as the proximity of the phone holder to a section of the road and is positively related to the likelihood that the user is in a dangerous situation. Its value is calculated as follows:

$$proximity\_level=\frac{direction\_matching}{avg\_distance}+\frac{1}{segment\_level}\ . \tag{6}$$

For the evaluation of *proximity_level*, we use several parameters, including *avg_distance*, *direction_matching*, and *segment_level*. *avg_distance* represents the average distance between the phone holder's position track and the road segment. *direction_matching* refers to the matching degree between the phone holder's position track and the direction of the road segment. *segment_level* is the product of the holder's current position point and the absolute distance of the specific road segment and the accuracy information of the current position point.

3.4 Interaction mode awareness algorithm

The awareness of the interaction between the phone holder and the mobile phone is another basic sub-problem of dangerous situation recognition. The interaction mode between them can directly reflect whether it is in the process of using the mobile phone. The algorithm for the interaction mode awareness can be designed and carried out in two aspects: mobile event and APP usage information.

Awareness based on interaction with mobile phone events. Through the system-level API provided by the mobile operating system (Android), we can effectively capture mobile events, including tapping the screen, sliding on the screen, using the soft keyboard for input, and locking or unlocking the phone. At the same time, as the dangerous situation recognition and prevention needs to have good real-time performance, so the capture of touchscreen events on mobile phones is periodic. Specifically, we use Δt to define the time interval of a cycle, using α_t and f_t respectively to indicate the number of touch events and the touch frequency captured during the time interval [t, t + Δt] (i.e., within one cycle from time t), here we can have:

$$f_t = \frac{\alpha_t}{\Delta t} \ . \tag{7}$$

At the same time, considering that the user starts browsing after opening the page without further touch operation, by using the touch frequency in the previous cycle, we could adjust the touch frequency in the current cycle, which could be represented as equation (8).

$$f_t^* = f_t + w * f_{t-\Delta t} \ . \tag{8}$$

Where w is the weight of the previous period touch frequency in the touch frequency correction value of this period (takes value in [0, 1)).

Besides, we can also capture the information of the phone holder's lock screen or unlock the screen by subscribing to the system broadcast to determine whether interactive mode awareness is required.

Interactive mode awareness based on mobile phone APP using information-assisted. Through the system-level API provided by the mobile operating system (Android), we can get the list of APPs currently used by the phone holder and the latest usage time of each app in the list. Based on this information, we define the APP usage status vector $\vec{\beta}$ in equation (9).

$$\vec{\beta} = (\beta_0, \beta_1, \ldots, \beta_{N-1}) \ . \tag{9}$$

Where N is the number of recently used apps by the phone holder recorded by the system, and β_i indicates the timestamp of the last time the phone holder used the i-th app.

For the APP use state vector $\vec{\beta}_t$ at time t and the APP use state vector $\overrightarrow{\beta_{t\text{-}\Delta t}}$ at time t - Δt, we can use two ways to describe the distance between vectors, and define them as d_1 in (10) and d_2 in (11).

$$d_1 = \left| len\left(\overrightarrow{\beta_{t\text{-}\Delta t}}\right) - len(\vec{\beta}_t) \right| \ . \tag{10}$$

$$d_2 = \frac{1}{2} \sum_{i=0}^{min\left(len\left(\overrightarrow{\beta_{t\text{-}\Delta t}}\right),\, len\left(\vec{\beta}_t\right)\right) - 1} \left(-sign\left(- \left| \vec{\beta}_t^{\,i} - \overrightarrow{\beta_{t\text{-}\Delta t}}^{\,i} \right| + 1\right) \ . \tag{11}$$

$$d = d_1 + d_2 \ . \tag{12}$$

Among them, d_1 represents the difference in dimension between APP usage state vectors at two moments, which means the difference in the number of APPs that the phone holder is using at two moments. d_2 represents the difference in the same dimension on the APP usage state vector at two moments, which means the difference in the last use time of the same APP at two moments. By inputting the difference into the *sign* function and deforming it, the number of apps that changed in the last usage time is finally counted. The distance of the APP usage state vector between two adjacent moments to obtain the APP usage change information is defined by the sum of d_1 and d_2 in equation (12).

In summary, we define the calculation of the interactive mode aware algorithm as equation (13).

$$interaction_{level} = (d_1 + d_2 + b) * \left(\left(f_t + w * f_{t\text{-}\Delta t}\right) + b\right) \ . \tag{13}$$

Among them, the purpose of introducing a constant term b (a smaller number greater than 0) is to prevent the result of the other side from being eliminated when one of the two methods is 0, and subsequent experiments will be based on this definition.

3.5 Integration of algorithms

Based on the algorithm designed for the three sub-problems, we could integrate these algorithms as well as design the structure of the mobile application, which is shown in **Figure 1**. The architecture of the research includes four layers, which are data sensing, dangerous context awareness and recognition, and the application layers. Each layer contains its function, and the algorithms are mainly integrated then used for context awareness and recognition. Our application is based on the architecture displayed in **Figure 1.**

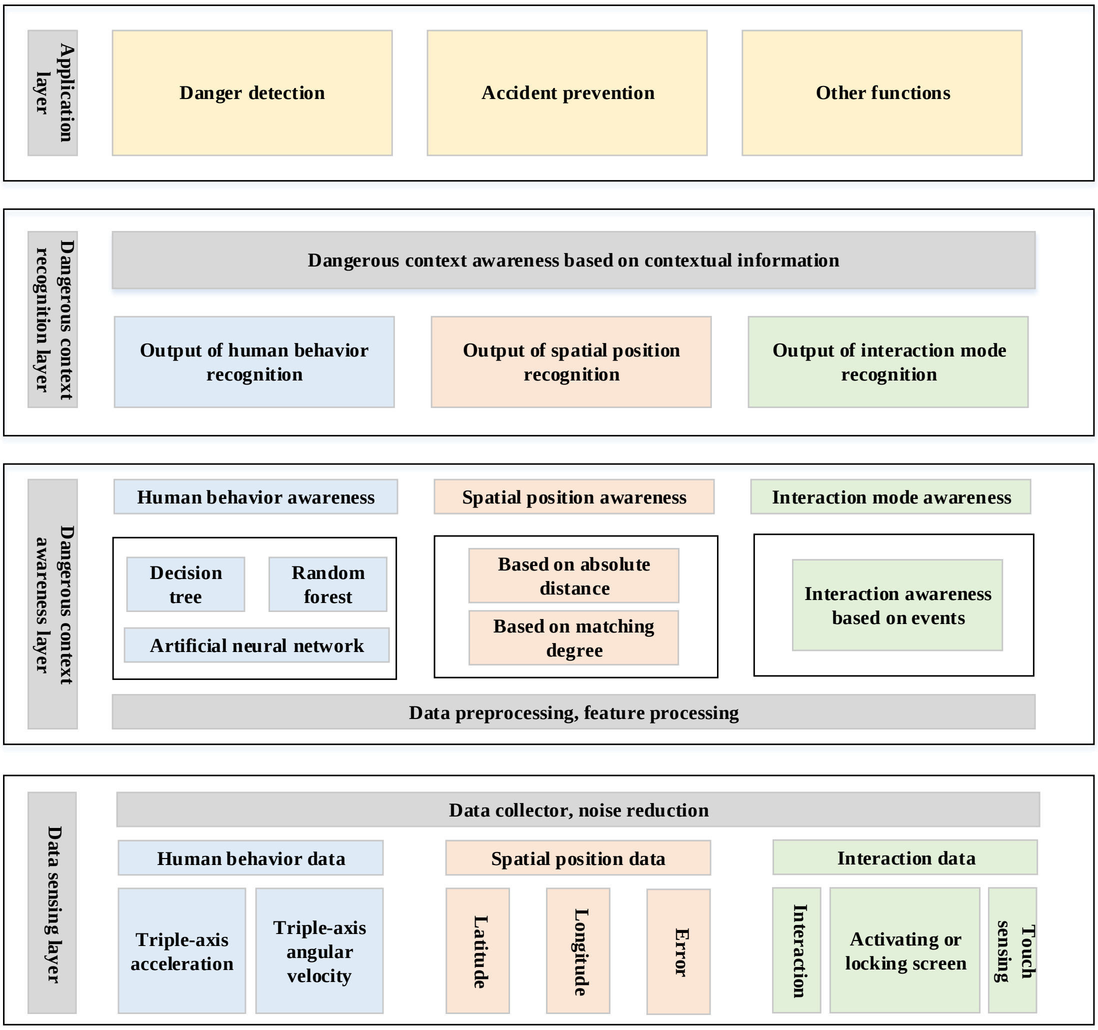
Application layer
Danger detection
Accident prevention
Other functions
Dangerous context recognition layer
Dangerous context awareness based on contextual information
Output of human behavior recognition
Output of spatial position recognition
Output of interaction mode recognition
Dangerous context awareness layer
Human behavior awareness
Spatial position awareness
Interaction mode awareness
Decision tree
Random forest
Artificial neural network
Based on absolute distance
Based on matching degree
Interaction awareness based on events
Data preprocessing, feature processing
Data sensing layer
Data collector, noise reduction
Human behavior data
Spatial position data
Interaction data
Triple-axis acceleration
Triple-axis angular velocity
Latitude
Longitude
Error
Interaction
Activating or locking screen
Touch sensing

Fig. 1. Architecture Diagram of Research in Accident Prevention During Mobile Phone Use Based on Context Awareness

4 Experiments and analysis

Based on the model and algorithms we designed, we carried out different control experiments using the test data we collected, the implementation and the results are shown in this part.

4.1 The human behavior recognition algorithm

In the experiment of the random forest-based human behavior awareness algorithm, several sets of experiments were set for the parameters *n_trees* and *n_feature_used*. In the control experiment, 2/3 of the feature dataset was used as the training set, and 1/3 was used as the test set. Both the training set and the test set are sampled from the feature dataset to ensure that both the training set and the test set have the same data distribution as the feature dataset.

The experimental results of parameter *n_trees* are shown in **Figure 2** and **Table 3** for details. During the experiments, parameter *n_feature_used* was kept with 20 percent of the original feature set. Like the curve in the figure shows, with the increase of *n_trees*, the misclassification rate of the classification goes down. Random forest is a kind of integrated learning model, which could improve the overall classification accuracy by training *n_trees* number of different decision tree classifiers (different in most case), and integrating the classification results to get voted. The correlation curve of *n_trees* and the misclassification rate is following this characteristic.

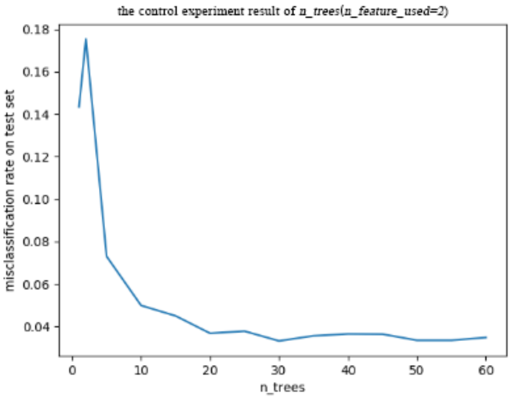

Fig. 2. Control experiment result of *n_trees*, the misclassification rate generally goes down with the increase of *n_trees*.

Table 3 Experiment result of *n_trees*.

n_trees	Misclassification rate	n_trees	Misclassification rate
1	0.1434	30	0.0356
2	0.0730	35	0.0365
5	0.0499	40	0.0364
10	0.0450	45	0.0335
15	0.0368	50	0.0335
20	0.0378	55	0.0335
25	0.0332	60	0.0348

Similarly, **Figure 3** and **Table 4** shows the experiment results of *n_feature_used*, and this time *n_trees* was kept in the value of 30. In this paper, we define *n_feature_used* as the ratio of the size of the feature set used in training the decision tree in the random forest to that of the original feature set. As the curve shows, the misclassification rate goes up with the increase of *n_feature_used*. One of the merits of random forests model is that each decision tree uses different subsets of features randomly sampled from the original feature set for training. As a result, each decision tree classifier used in classification tends to use different attributes, and as *n_feature_used* increases, the probability of multiple decision tree classifiers in random forests using feature subsets with high repetition rate has increased, which further improves the performance of integrated multi-classification of random forest.

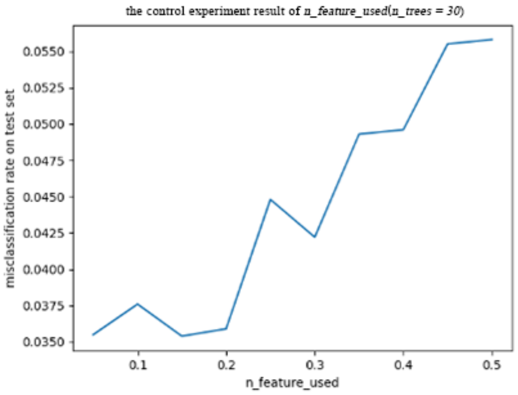

Fig. 3. Control experiment result of *n_feature_used*, the misclassification rate generally goes up with the increase of *n_trees*.

Table 4 Experiment result of *n_feature_used*.

n_feature_used	Misclassification rate	*n_feature_used*	Misclassification rate
0.05	0.0355	0.30	0.0422
0.10	0.0376	0.35	0.0493
0.15	0.0354	0.40	0.0496
0.20	0.0359	0.45	0.0555
0.25	0.0448	0.50	0.0558

According to the experimental results, *n_trees*=30 and *n_feature_used* = 0.2 were used as the preferred parameters of human behavior awareness model based on the random forest. Under these parameters, the model showed 96.47% accuracy on the test set. Due to the large number of classifications, we use the confusion matrix shown in **Figure 4(a)** to display the results.

At the same time, this paper compares the results of the random forest algorithm and the artificial neural network algorithm used in the implementation of the human behavior recognition algorithm. It contains 9 input layer neurons, 25 hidden layer neurons, and 32 output layer neurons. The learning rate is 0.05 and ReLU is used as the activation function, and we perform 400,000 times of training. The obtained recognition results are compared in the confusion matrix as shown in **Figure 4(b)**.

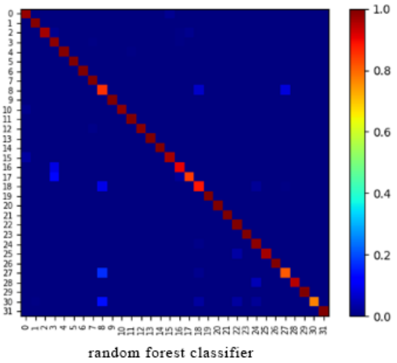

Fig. 4(a). The confusion matrix of the random forest classifier, the classifier performs well on the vast majority of the test data.

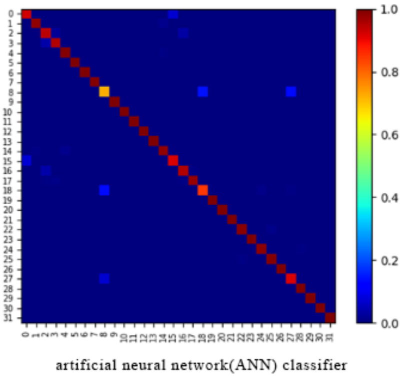

Fig. 4(b). The confusion matrix of the artificial neural network(ANN) classifier, though it has a good performance, yet it has more errors than the random forest classifier in general, and it demands more resources and time

As **Figure 4** shows, the random forest and ANN classifier both have a reliable accuracy, and the former performs well generally. Since the application scenario of this paper is a smartphone, which has the characteristics of more precious computing resources, and because the random forest classification model has high accuracy and can effectively run on a large data set. Considering the accuracy as well as the resources and time factor, we use the random forest as the classification model to solve the human behavior recognition problem.

4.2 Spatial position awareness algorithm & Interaction mode awareness algorithm

By setting up a recording experiment to verify the effect of the spatial position sensing algorithm implemented in this chapter, the sampling frequency of the position trajectory of all mobile phone holders is set at 0.2 Hz. According to the evaluation method defined above, the highest matching degree of the current position trajectory of the mobile phone holder is 4481.2666.

Carrying out the second group and the third group of experiments in different situations, the highest matching degree of the user's current position trajectory are 2352.9992 and 14332.1282 respectively. The experimental results show that the algorithm implemented in this paper based on the current position of the mobile phone holder and the past data, as well as the historical position trajectory over some time, can effectively determine how close the mobile phone holder is to a specific road segment at this time.

Figure 5 shows the experimental results of the interactive mode sensing algorithm when both *w* and *b* are set to 0.5 in equation (13). The experimental results show that the interactive mode sensing algorithm designed and implemented in this chapter can better judge whether the mobile phone holder is in the process of using the phone or not.

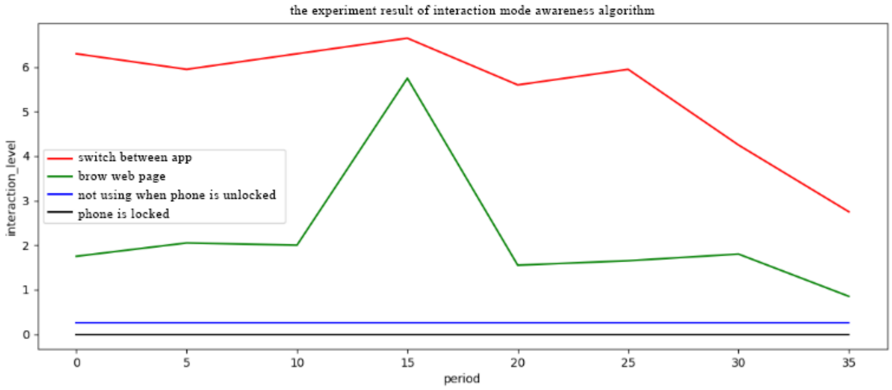

Fig. 5. The experiment result of interaction mode awareness algorithm. The interaction level of switching between the apps is the highest, followed by browsing web pages, not using the phone and when the phone is locked

5 Application

After designing and implementing the corresponding algorithms for the three sub-problems of dangerous situation identification, this chapter designs and implements the dangerous situation prevention system based on the dangerous situation recognition model and the sub-question model. Specifically, we complete the user management module and data acquisition module and the dangerous situation identification and prevention module. The functions of the dangerous situation prevention system design include user management, data collection and management, dangerous situation identification and prevention (including real-time identification of three sub-problems and prompting information according to the risk level, terminating the interaction or forced shutdown, and so on).

6 Conclusion and future work

This paper proposes a method and system for preventing accidental injuries during the use of mobile phones. The research is based on context-awareness, and we define the dangerous situation that mobile phone holders may come into when using the mobile phone from three aspects: human behavior, spatial location, and the interaction with the mobile phone. The paper also proposes the research methods of dangerous situation recognition and the dangerous situation recognition model. Based on this model, this paper implements a dangerous situation recognition and prevention system (Android application). For the study of dangerous situation identification and prevention, the follow-up work will be improved based on the following points. For the formal definition of dangerous situations, we will try to identify dangerous situations from more angles to further increase the accuracy of the dangerous situation identification results. Different dangerous situation level will be distinguished to better intervene in the user's use of mobile phones in dangerous situations. To make our research have practical significance, we will define more specific dangerous situations to match the situations in people's daily work and life, and further improve the dangerous situation identification prevention model. For human behavior awareness sub-problems, we will try to improve the definition and data collection of variables including behavior patterns, mobile phone position, mobile phone direction orientation and so on. For spatial location awareness sub-problems, we will try to define more dangerous areas besides traffic sections, as well as explore the research methods of whether the mobile phone holder is in a dangerous area. For the interactive mode awareness sub-question, much work will be done to judge whether the mobile phone holder is holding the mobile phone from more angles, such as the light intensity given by the light sensor, and whether the system is playing a sound an so forth.

Preferences

[1] World Health Organization. Global status report on road safety 2015[M]. World Health Organization, 2015. MLA

[2] Abowd G, Dey A, Brown P, et al.: Towards a better understanding of context and context-awareness[C]// Handheld and ubiquitous computing. Springer Ber-lin/Heidelberg. 304-307 (1999)

[3] Bangerter B, Talwar S, Arefi R, et al.: Networks and devices for the 5G era[J]. IEEE Communications Magazine (2014)

[4] Shin S H, Lee M S, Park C G, et al.: Pedestrian dead reckoning system with phone location awareness algorithm[C]// Position Location and Navigation Symposium (PLANS), 2010 IEEE/ION. IEEE, 97-101 (2010)

[5] Josu é Iglesias, Cano J, Bernardos A M, et al.: A ubiquitous activity-monitor to prevent sedentariness[C]// IEEE International Conference on Pervasive Computing & Communications Workshops. IEEE (2011)

[6] Jeon C K, Kim N H, In H P. Situation-awareness overheating protection solution for mobile devices[C]// IEEE International Conference on Consumer Electronics. IEEE (2015)

[7] Zaldivar J, Calafate C T, Cano J C, et al.: Providing accident detection in vehicular networks through OBD-II devices and Android-based smartphones[C]// Local Computer Networks. IEEE (2011)

[8] Anil B S, Vilas K A, Jagtap S R: Intelligent system for vehicular accident detection and notification[C]// International Conference on Communications & Signal Processing. IEEE (2014)

[9] Fabio Martinelli, Francesco Mercaldo, Vittoria Nardone, et al.: Context-Awareness Mobile Devices for Traffic Incident Prevention[C]// International Conference on Pervasive Computing and Communications Workshops. IEEE (2018)

[10] Haque A, Kode S, Razaque A, et al.: ConSec: An encryption policy for context aware security applications[C]// International Conference on Information & Communication Systems. IEEE (2016)

[11] Ong Y J, Qiao M, Routray R, et al.: Context-Aware Data Loss Prevention for Cloud Storage Services[C]// IEEE International Conference on Cloud Computing. IEEE (2017)

[12] Kleinberger T, Becker M, Ras E, et al.: Ambient Intelligence in Assisted Living: Enable Elderly People to Handle Future Interfaces.[J]. (2007)