2011 23rd IEEE International Conference on Tools with Artificial Intelligence

Using Artificial Neural Network To Determine Favorable Wheelchair Tilt and Recline Usage In People With Spinal Cord Injury

Training ANN with Genetic Algorithm to Improve Generalization

Jicheng Fu, Jerrad Genson
Department of Computer Science
University of Central Oklahoma
Edmond, OK, USA
{jfu, jgenson} @uco.edu

Yih-Kuen Jan, Maria Jones
Department of Rehabilitation Sciences
University of Oklahoma Health Sciences Center
Oklahoma City, OK, USA
{Yih-Kuen-Jan, Maria-Jones} @ouhsc.edu

Abstract— **People with spinal cord injury (SCI) are at risk for pressure ulcers because of their poor motor function and consequent prolonged sitting in wheelchairs. The current clinical practice typically uses the wheelchair tilt and recline to attain specific seating angles (sitting postures) to reduce seating pressure in order to prevent pressure ulcers. The rationale is to allow the development of reactive hyperemia to re-perfuse the ischemic tissues. However, our study reveals that a particular tilt and recline setting may result in a significant increase of skin perfusion for one person with SCI, but may cause neutral or even negative effect on another person. Therefore, an individualized guidance on wheelchair tilt and recline usage is desirable in people with various levels of SCI. In this study, we intend to demonstrate the feasibility of using machine-learning techniques to classify and predict favorable wheelchair tilt and recline settings for individual wheelchair users with SCI. Specifically, we use artificial neural networks (ANNs) to classify whether a given tilt and recline setting would cause a positive, neutral, or negative skin perfusion response. The challenge, however, is that ANN is prone to overfitting, a situation in which ANN can perfectly classify the existing data while cannot correctly classify new (unseen) data. We investigate using the genetic algorithm (GA) to train ANN to reduce the chance of converging on local optima and improve the generalization capability of classifying unseen data. Our experimental results indicate that the GA-based ANN significantly improves the generalization ability and outperforms the traditional statistical approach and other commonly used classification techniques, such as BP-based ANN and support vector machine (SVM). To the best of our knowledge, there are no such intelligent systems available now. Our research fills in the gap in existing evidence.**

Keywords - Pressure Ulcer; Skin Perfusion; Wheelchair Tilt and Recline; Artificial Neural Network; Support Vector Machine; Genetic Algorithm

I. INTRODUCTION

Pressure ulcer has been identified as the most common complication for people with spinal cord injury (SCI) [1, 2]. It significantly affects the quality of life and overall healthcare costs of wheelchair users with SCI. It is estimated that more than half of the people with SCI will develop at least one pressure ulcer in their lifetimes [3]. The estimated annual cost on the treatment of pressure ulcers in people with SCI amounts to approximately 1.4 billion dollars, accounting for 25% of the total cost of treating SCI [4].

The essential component of a pressure ulcer prevention program is to periodically reduce seating pressure [5]. During the pressure-relieving period, ischemic soft tissues will be able to restore blood flow to meet the metabolic needs of local cells. Inadequate pressure relief will cause irreversible damage in ischemic tissues. Clinically, the pressure relieving activities performed by wheelchair tilt (a change of seat angle orientation while maintaining the seat-to-back angle) and recline (a change of the seat-to-back angle) is recommended for preventing sitting-induced pressure in individuals with SCI, especially in quadriplegia [6, 7]. The principle of wheelchair tilt and recline usage is based on the evidence that turning the patient every 2 hours results in a lower incidence of pressure ulcers [8]. The purpose of periodically performing wheelchair tilt and recline is to allow the development of reactive hyperemia to re-perfuse the ischemic tissues [9].

On the other hand, it is unclear at what angles the wheelchair tilt and recline usage would provide adequate pressure relief for enhancing skin blood flow. Typically, the current clinical practice provides uniform guidance to all wheelchair users with SCI. However, our study [10] indicates that the skin blood flow response to wheelchair tilt and recline usage varies largely from person to person with SCI. Therefore, it is desirable to develop an intelligent system that can provide individualized guidance on wheelchair tilt and recline usages.

Artificial neural network (ANN) is a powerful computational model with many appealing properties, such as learning capability, adaptability, and ability to generalize [11]. In [10], we used the back-propagation algorithm (BP) [12] to train ANN to perform binary classification, i.e., the skin blood flow response to wheelchair tilt and recline settings is classified to be either positive (i.e., favorable for skin perfusion increase) or negative. The results show that ANN significantly outperforms the traditional classical approach in correctly classifying wheelchair tilt and recline settings. In the subsequent study, however, we realize that binary classification may increase chances of false positive and/or false negative predictions. The major reason is that skin perfusion may only marginally increase or decrease in

1082-3409/11 $26.00 © 2011 IEEE
DOI 10.1109/ICTAI.2011.13

IEEE computer society

some cases. Factors, such as measurement precision, noise, operation issues, etc., may contribute to such kinds of marginal variations. Hence, it is inappropriate to classify such marginal cases to be either positive or negative.

We propose to use ANN to perform ternary classification, i.e., classify the blood flow response to wheelchair tilt and recline settings into three classes, namely, positive, neutral, and negative. In this study, we first use the BP algorithm to train ANN. Although this approach perfectly classifies the existing skin perfusion data, it generalizes very poorly. In other words, BP-based ANN is prone to overfitting, in which ANN perfectly classifies the existing data but fails to classify new (i.e., unseen) data.

To improve ANN's generalization capability, we use the genetic algorithm (GA) [13] to train ANN to reduce chances of being stuck in a local optimum. GA is a population-based stochastic search approach that has been widely applied in various research areas. Our implementation of GA absorbs many techniques from existing research results [14, 15] to diversify the population and improve the solution quality. Our research results indicate that GA-based ANN significantly outperforms BP-based ANN, the traditional statistical approach, and other commonly used classification techniques, such as support vector machine (SVM), in generalization.

To the best of our knowledge, this is the first study that aims to perform ternary classification so that individualized wheelchair tilt and recline guidance can be provided to people with SCI. The goal of this study is to demonstrate the feasibility of using machine-learning techniques to construct such an intelligent model.

The rest of the paper is organized as follows. In Section II, we introduce background knowledge of ANN and GA. In Section III, we present the experiment that we performed to investigate skin perfusion response to wheelchair tilt and recline usages. In Section IV, we discuss approaches that perform binary classifications and related issues. In Section V, we present approaches for ternary classifications to overcome the issue of binary classification. Then, we show the experimental results of ternary classification in Section VI, present the discussion in Section VII, and conclude in Section VIII.

II. Background

In this section, we introduce the major techniques used in this study.

A. *Artificial Neural Network* (*ANN*)

An ANN can be defined as a set of simple processing units (neurons) that communicate among themselves by sending signals. The signals travel through weighted connections between neurons. Upon receiving signals, these neurons accumulate the inputs and produce outputs according to their internal activation functions. The outputs can serve as inputs for other neurons, or can be a part of the network outputs [11]. More precisely, we use multilayer perceptron (MLP) [16], in which neurons are organized into ordered layers. Connections between neurons are allowed only between adjacent layers to accept inputs and send outputs. As shown in Figure 1, each circle represents a neuron. From left to right, the first layer is called the input layer, the second layer is called the hidden layer, and the third layer is called the output layer.

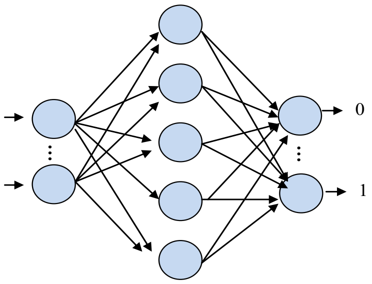

Figure 1. An Example of ANN

The commonly used algorithm to train ANN is the back-propagation algorithm (BP) [12]. Specifically, the output errors are measured by,

$$E = \sum_{p=1}^{P} \sum_{i=1}^{S} (v_i^p - o_i^p)^2 \qquad (1)$$

where P is number of training data and S is the number of units in the output layer; v_i^p is the desired output value and o_i^p is the actual output. The goal is to minimize E. To do so, the gradient descent rule is applied to E, i.e., $\frac{\partial E}{\partial w_{ij}}$ for all weights w_{ij} such that the weights are adjusted in the descending gradient direction, which is shown as follows:

$$w_{ij} = w_{ij} - \eta \frac{\partial E}{\partial w_{ij}} \qquad (2)$$

where $\eta > 0$ is a constant representing the learning rate. The above calculation is repeatedly performed until the error E drops to meet a stop criterion or a predefined iteration number has been reached.

Nevertheless, the BP algorithm is prone to converge on local optima. As an alternative, the genetic algorithm (GA) [13] can be used to train ANN [11].

B. *Genetic Algorithm* (*GA*)

```
1. Create the initial population of randomly generated
   chromosomes
2. Perform selection on the population (selection operator).
3. Perform crossover on the remaining chromosomes to
   produce child population (crossover operator).
4. If the max generation has been exceeded, return the fittest
   chromosome.
5. If any chromosome is greater than or equal to the minimum
   fitness threshold
6.     return the chromosome
7. Otherwise, return to step 2.
```

Figure 2. Genetic Algorithm Outline

Figure 2 illustrates the major steps of genetic algorithm (GA), which is a population-based stochastic optimization search approach. GA begins with a randomly generated population (line 1). A fitness function is used to evaluate the population. Based on the fitness values, GA updates the population and searches for optimal solutions using randomization techniques inspired by natural evolution, such as mutation, selection, crossover, etc.

III. Experiment

We performed a study to investigate skin blood flow response to wheelchair tilt and recline settings at the Biomechanics and Microcirculation Laboratory at the University of Oklahoma Health Sciences Center [17]. Eleven adult subjects with SCI were recruited to participate in the study. Two of the subjects were women and nine of the subjects were men. The mean standard deviation age was 37.7 ± 14.2 years and the duration of injury was 8.1 ± 7.5 years. Inclusion criteria included traumatic SCI at the level of C4 through T12, at least 6 months after spinal injury, and use of a wheelchair as a primary means of mobility.

A. Instrument

Laser Doppler flowmetry (LDF) (Periflux System 5001, Perimed, Sweden) was used to measure skin blood flow (mlLDF/min/100g tissue). Two thin and flexible probes (PR415 probe, Perimed) were used to measure skin blood flow over the skin on the right ischial tuberosity and spinal process of the sacrum (midline between two posterior superior iliac spines (PSIS)). Skin blood flow over the ischial tuberosity was the primary measurement and the sacral area was the reference for assessing wheelchair tilt and recline usage on lower back tissue viability. LDF provided noninvasive measurement of skin blood flow at a depth of about 1 mm via laser and fiber optics technology. A low power beam (1 mW) of solid-state diode laser (780 nm wavelength) was delivered to the skin. The blood flow signal was sampled at 20 Hz to fully characterize blood flow oscillations.

B. Factorial Study Design

A repeated measures factorial design was performed to determine the effects of wheelchair tilt and recline settings on skin blood flow response to sitting-induced pressure. We compared skin blood flow response (i.e., skin perfusion) to common clinical recommendations of tilt and recline usage. The main factors include tilt angles at 15°, 25°, and 35° and recline angles at 100° and 120°. A combination of 3 tilt and 2 recline angles resulted in 6 testing conditions. The order of the 6 testing conditions was randomly assigned to the subjects. Each condition started with a baseline period (i.e., sitting-induced ischemic period with no tilt and recline) for 5 minutes, followed by the pressure relieving period for another 5 minutes. In addition, the subject assumed a sitting posture of 35 degree tilt and 120 degree recline for a duration of 5 minutes to restore blood flow supply to ischemic tissues between each conditions [18]. Each subject spent about 90 minutes to complete the experimental protocol.

The skin perfusion b_0 was measured during the first 5-minute, i.e., the ischemic period. Then, the skin perfusion b_1 was measured during the next 5-minute which was the pressure reduction period. The skin perfusion increase was computed by the ratio as follows:

$$\beta = b_1 / b_0 \quad (3)$$

A 5-minute tilt and recline position is sufficient for a full recovery of skin perfusion because a 5-minute sitting-induced ischemia was used in this study.

C. Attributes of Participants

In [10], we studied 5 attributes of participants, including 2 demographic attributes, namely, age and gender, and 3 neurological attributes, namely, level of injury, duration of injury, and completeness. All these attributes are reported to be risk factors for pressure ulcers [2, 19]. We use the same set of attributes in this study because the main purpose is to demonstrate the feasibility of using machine-learning techniques to classify whether a given wheelchair tilt and recline setting is favorable for a wheelchair user with SCI. We will address the issue of identifying attributes relevant to skin perfusion in our future study.

IV. Binary Classification

In [17], we used the average skin perfusion $\bar{\beta}$ in each tilt and recline setting to classify the skin perfusion data in the same setting. Specifically, if $\bar{\beta} > 1$, then we classify all the data in that setting to be positive. Otherwise, we classify data to be negative. The advantage of this approach is its simplicity. We did find a pattern from the average skin perfusion $\bar{\beta}$: as the angles of tilt and recline increase, the average skin perfusion $\bar{\beta}$ also increases. The disadvantage is, however, the classification accuracy is low -- only 59.38%.

To improve the classification accuracy, we used artificial neural network (ANN) to classify skin perfusion data based on the 5-attributes above [10]. As shown in Table I, two experiments were performed to evaluate the classification and generalization ability of the learned model. "Train and test with the same set" means that we trained and tested ANN with the same set of data. ANN classified almost all the data correctly. However, with a small data set, overfitting can easily happen. The 10-fold cross-validation is a commonly used approach to mitigate the overfitting impact [20]. 10-fold cross-validation *randomly* divides the training data into 10 equal and mutually exclusive sets (folds). Each time, the approach chooses one fold to be testing data and combines the other 9 folds as training data. Therefore, ANN is always tested with unseen data. The results obtained from the 10 folds are averaged to produce a single accuracy rate. The results (see row "ANN") in Table I show that overfitting did happen because the accuracy rate dropped from 96.88% (on train and test with the same set) to 71.83% (on 10-fold cross-validation).

TABLE I BINARY CLASSIFICATION

Classification Algorithm	**Experiments**	
	Train and Test with the Same Set	***10-fold Cross-Validation***
ANN	96.88%	71.83%
SVM	100%	78.13%

In addition, we also used another classification algorithm, namely, support vector machine (SVM), to classify the skin perfusion data. Unlike ANN, which is prone to be stuck in local optima [11], SVM always finds the global optimum [21]. The results in Table I show that SVM performed slightly better than ANN.

To further evaluate the generalization capability of the learned intelligent model, i.e., how well the learned model can classify unseen data, we performed a "leave-one-out" experiment. The idea is that we left out data associated with a participant as testing data and combined the other 10 participants' data as training data. In fact, the "leave-one-out" approach is an 11-fold cross-validation approach since 11 subjects participated in the study. The advantage of this approach over 10-fold cross-validation is that data are not randomly partitioned; instead, data in each fold belong to a particular participant. Therefore, ANN is always tested by an unseen participant and, hence, testing data are more meaningful.

TABLE II LEAVE-ONE-OUT EXPERIMENT

Classification Algorithm	**Leave-One-Out**
ANN	41.67%
SVM	73.48%

As shown in Table II, the accuracy rate of ANN dropped sharply to 41.67% on the leave-one-out experiment. In comparison, SVM still maintained satisfying accuracy rate. Due to this experimental result, we include SVM in this study as a baseline to compare with the GA-based ANN approach.

A. *Issues of Binary Classification*

In binary classification, we classify the skin perfusion data β (see (3)) as either negative or positive. Specifically, if $\beta > 1.0$, we classify it as positive (or favorable for skin perfusion). Otherwise, we classify it as negative. However, this classification scheme may report false positive and/or false negative if β is only marginally greater or less than 1.0. For example, one skin perfusion ratio in our study is $\beta = 1.034$, which means that the skin perfusion is only marginally increased. If we consider factors, such as the measurement precision, environment noise, etc., it is hard to tell whether this data is truly positive or not.

V. TERNARY CLASSIFICATION

Alternatively, it would make more sense if we classify the skin perfusion data into three classes, namely, negative, neutral, and positive. In this case, we need to determine a threshold τ such that

- a skin perfusion ratio β is classified as negative if and only if $\beta < 1.0 - \tau$;
- β is classified as neutral if and only if $1.0 - \tau \leq \beta \leq 1.0 + \tau$; and
- β is classified as positive if and only if $1.0 + \tau < \beta$.

Specifically, we intend to determine a function $f \to \{-1, 0, 1\}$, where 0, -1, and 1 represent neutral, negative, and positive skin perfusion increase, respectively; the parameters of f include age, duration of injury, gender, level of injury, completeness, and wheelchair tilt and recline angles.

We investigate three approaches to perform ternary classifications on skin perfusion data.

A. *Traditional Statistical Approach*

First, we use the traditional statistical approach to classify skin perfusion data based on the average skin perfusion ratio $\bar{\beta}$ in each tilt and recline setting. The idea is similar to that in binary classification. Data in the same tilt and recline setting are classified as negative if and only if $\bar{\beta} < 1.0 - \tau$. Similarly, data in the same setting are classified as neutral if and only if $1.0 - \tau \leq \bar{\beta} \leq 1.0 + \tau$ and, otherwise, data are classified as positive. Here, τ is the threshold facilitating the classification.

B. *Using BP to Train ANN*

Second, we use the back-propagation (BP) algorithm to train ANN to classify skin perfusion data. We run the MultiLayerPerceptron (i.e., ANN) in Weka [22] to conduct this experiment. We use the default settings of ANN parameters, which set the learning rate to be 0.3, momentum to be 0.2, and the iteration number to be 500. In addition, support vector machine (SVM) performed better than BP-based ANN on binary classification. Hence, we also use SVM with the kernel of polynomial to perform ternary classification so that we can compare whether SVM still outperforms BP-based ANN on ternary classification.

C. *Using GA to Train ANN*

Finally, we use the genetic algorithm (GA) to train ANN to improve the generalization capability. In order to use GA, we need to model the solution domain with a genetic representation, define the fitness function, and define GA operators such as selection, crossover, and mutation.

1) Genetic representation of the solution domain. Traditionally, a potential solution (i.e., a chromosome) is encoded in a binary string (i.e., 0's and 1's). The disadvantage of such an encoding is that it is not intuitive. In our study, we choose a natural encoding, which encodes a chromosome as a weight matrix of the ANN. Specifically, for a particular ANN, its architecture is fixed. The corresponding weights of the connections between neurons represent a solution. Therefore, a population of GA consists of individual chromosomes that are encoded as weight matrices.

2) Fitness function. Figure 3 shows how the fitness function is calculated. Given a weight matrix m which is a potential solution to ANN, the fitness function counts the correct classifications based on training data D. In this ternary classification scheme, we label sample data as "1" (i.e., positive), "0" (i.e., neutral), and "-1" (i.e., negative). For each data item d in the training set D, ANN calculates the output based on the given weight matrix m (line 2 and

3). If ANN correctly classifies *d* (line 4), the fitness value is increased by 1. Here, the function label(*d*) returns the label of data item *d*, i.e., -1, 0, or 1. If ANN misclassifies *d*, but the difference is 1, which means either the actual or the expected value is neutral (i.e., 0), then we say that this classification is only half-correct and the fitness value is increased by 0.5 (line 7). Otherwise, it is a complete misclassification and the fitness value will not be changed.

```
/*m is a weight matrix; D is the set of training data*/
Function Fitness (m, D)
1.  correct ← 0;
2.  for each data item d ∈ D do
3.      result ← ANN(m, d)
4.      if result == label(d) then
5.          correct ← correct + 1
6.      else if |result – label(d)| == 1 then
7.          correct ← correct + 0.5
8.      end if
9.  end do
10. fitness ← correct / sizeof(D)
11. return fitness
```

Figure 3. Fitness Function

3) Selection. To improve genetic diversity, we use tournament selection [14] as the selection algorithm. This algorithm selects chromosomes at random from the population to "compete" in a series of tournaments. The winner is determined according to the fitness value. This selection algorithm provides a chance for chromosomes that are not the absolute fittest. This algorithm is also relatively easy to implement in a multiprocessing architecture (i.e., worker processes are responsible for conducting tournaments on different CPU cores) to improve efficiency (see Section (V.C.7) for more information).

The number of chromosomes that are allowed to survive is determined by the adaptation operator, which is designed to improve the solution quality and increase the rate of convergence.

4) Adaptation operator. Inspired by adaptive population sizing schemes in [15], we propose our own adaptation operators, which consist of the mutation adaptation operator and the population adaptation operator. These adaptation operators compute a new mutation rate and population size based on the current median fitness of the gene pool. The basic idea is to start with a large population and mutation size (the "exploration phase") to increase solution quality, and gradually decrease the size of both the population and mutation (the "exploitation phase") to decrease the number of calculations that GA has to perform per iteration, causing it to converge faster.

Figure 4 shows the algorithms for the adaptation operators. There are two constants involved in the mutation adaptation operator to compute the new mutation ratio. MAX_MUT_RATIO and MIN_MUT_RATIO are maximum and minimum mutation ratios, which are set to be 0.8 and 0.09, respectively. Similarly, there are two constants involved in the population adaptation operator to compute the new population size. MAX_POP_DECREASE defines the maximum population decrease ratio, which is set to 0.75. MIN_POP_SIZE is the minimum population size, which is set to 25. Be noted that these constants are determined during the experiments in this pilot study. The issue of determining the optimal values of these constants will be reported in the future study.

```
Function Mutation_Adaptation_Operator (median_fitness)
    reduction_ratio ← 1.0 – median_fitness
    new_mut_ratio ← MAX_MUT_RATIO × reduction_ratio
    If new_mut_ratio < MIN_MUT_RATIO then
        return MIN_MUT_RATIO
    else
        return new_mut_ratio
    end if
end Function

Function Population_Adaptation_Operator (median_fitness,
old_pop_size)
    reduction_ratio ← 1.0 – median_fitness ×
    MAX_POP_DECREASE
    new_pop_size ← ⌈old_pop_size × reduction_ratio⌉
    if new_pop_size < MIN_POP_SIZE then
        return MIN_POP_SIZE
    else
        return new_pop_size
    end if
end Function
```

Figure 4. Adaptation Operators

5) Crossover. For crossover, we apply the commonly used approach, i.e., take two chromosomes, namely, ch_1 and ch_2, and combine the first half of ch_1 with the second half of ch_2 to produce a new chromosome. Every chromosome is crossed with other chromosomes. Hence, the resulting child population size will be the square of the post-selection parent population.

6) Mutation. The mutation operator accepts two parameters, namely, *probability* and *weights*, where *probability* is the probability that a weight might be mutated, and *weights* is a matrix of the connection weights that can be used to configure ANN. Each weight is iterated over with a probability that it might be mutated. The magnitude of the mutation is calculated with a function that randomly returns values from the range of -3.0 and +3.0. The boundary of -3.0 to 3.0 is determined from the weights of ANN obtained by running the MultiLayerPerceptron function of Weka.

7) Other optimizations. We perform two additional optimizations to improve the solution quality and the problem solving efficiency. First, we use the technique of "seed chromosome" to improve the solution quality. Specifically, we run BP-based ANN in Weka on each fold to obtain a weight matrix. This weight matrix is used as one of the chromosomes in the initial population. The purpose is to ensure that the learning quality of GA-based ANN is at least as good as that of BP-based ANN. Second, we implement the GA algorithm in Python. Hence, we can take advantage of Python's built-in multiprocessing capability to spawn one worker process for each available CPU core. Specifically, the controller process communicates with worker processes via two multiprocessing queues, namely, an inbound queue and an outbound queue. When the controller puts an object onto the outbound queue, only one worker process can pop it off. It does not matter which worker process receives the object because all workers are identical. When a worker finishes processing an object, it pushes it onto the inbound queue, and the controller process pops it off. As mentioned previously, the selection and crossover operations can benefit from such a multi-processing mechanism as well.

VI. Results of Ternary Classification

In this study, we examined two thresholds, namely, $\tau = 0.10$ and $\tau = 0.15$. Table III shows the classification accuracy for the traditional approach that used the average skin perfusion ratio $\bar{\beta}$ to classify data in the same tilt and recline setting. Compared to the result of binary classification (i.e., 59.38%), the use of average data to perform ternary classification led to lower classification accuracy.

Table III Traditional statistical approach

Threshold	Classification Accuracy
$\tau = 0.10$	46.97%
$\tau = 0.15$	48.48%

Since the traditional approach cannot accurately classify the skin perfusion data, we used the BP algorithm to train ANN to classify skin perfusion data. As shown in Table IV, ANN could perfectly classify existing data (see column "train and test with the same set"), which is a significant improvement over the traditional approach.

However, overfitting did happen because the averaged classification accuracy on "leave-one-out" dropped dramatically. As a baseline, we also used SVM to perform the same experiments. The classification accuracy of SVM was slightly better than that of ANN.

Table IV Results of Using BP to Train ANN

Algorithm	Experiments		
	Threshold **(*τ*)**	***Train and Test with the Same Set***	***Leave-One-Out***
BP_ANN	0.10	100%	28.03%
BP_ANN	0.15	100%	21.21%
SVM	0.10	100%	36.58%
SVM	0.15	100%	49.24%

Since using BP to train ANN did not show satisfying generalization ability, we examined whether using GA could improve the generalization ability of ANN. Since the BP-based ANN performed well on train and test with the same set as shown in Table IV, there is no need to use GA to train ANN based on the whole data set. Instead, we used GA to train ANN on the "leave-one-out" experiment.

Table V Detailed Results on Leave-One-Out Experiment

Folds	BP-ANN		GA-ANN	
	τ= 0.10	**τ= 0.15**	**τ= 0.10**	**τ= 0.15**
1	16.67%	16.67%	100.00%	50.00%
2	0	16.67%	16.67%	16.67%
3	0	16.67%	16.67%	50.00%
4	25.00%	0	25%	25.00%
5	66.67%	66.67%	66.67%	66.67%
6	100.00%	83.33%	100.00%	83.33%
7	0	16.67%	66.67%	50.00%
8	50.00%	0	66.67%	33.33%
9	16.67%	0	66.67%	50.00%
10	0	0	100.00%	66.67%
11	33.33%	16.67%	100.00%	50.00%
Average	28.03%	21.21%	65.91%	49.24%

The experiment results in Table V show that using GA to train ANN significantly improves the generalization capability. The average classification accuracy is more than doubled on both threshold categories, i.e., from 28.03% to 65.91% (with $\tau = 0.10$) and from 21.21% to 49.24% (with $\tau = 0.15$). In addition, GA-based ANN also performs significantly better than SVM (see rows of "SVM" in Table IV) with $\tau = 0.10$.

VII. Discussion

First of all, we demonstrate that it is feasible to use machine-learning techniques to classify favorable tilt and recline settings for individual wheelchair users with SCI. When using the BP algorithm to train ANN, it perfectly classified the existing data (see column "train and test with the same set" in Table IV). In comparison, the traditional approach that used the average skin perfusion data to perform classifications only achieved an accuracy of 59.38% on binary classification, 46.97% on ternary classification with $\tau = 0.10$, and 48.48% with $\tau = 0.15$.

Second, the objective of this study is to answer the question -- "Is a given wheelchair tilt and recline setting favorable for skin perfusion increase for a particular wheelchair user with SCI?” To answer this question, we investigate different approaches that classify the skin perfusion data into two classes (i.e., binary classification) and into three classes (i.e., ternary classification). We then point out that ternary classification is more advantageous than binary classification in that it can reduce the chances of false positive/negative cases, in which the skin perfusion increase ratio β (see (3)) is only marginally greater or less than 1.0. For ternary classification, we examined two thresholds, namely, 0.10 and 0.15. The experimental results show that when using the threshold of 0.10, BP-based ANN and GA-based ANN performed better in classifying favorable tilt and recline settings than when using the threshold of 0.15. Therefore, the threshold of 0.15 may be too big to truly reduce false positive and negative classifications.

Third, with a small data set (e.g., there were only 11 participants in this study), overfitting is likely to happen. This is the major challenge for ternary classification. To minimize the overfitting impact and examine the generalization ability of the trained ANN, we performed “leave-one-out” experiments that left out data associated with a participant as testing data and combined the rest of data as training data. The results show that overfitting did happen: BP-based ANN only achieved a classification accuracy of 28.03% with $\tau = 0.10$ and 21.21% with $\tau = 0.15$. This initiated the need to use GA to train ANN to avoid local optima and achieve better generalization ability. Our experimental results show that using GA to train ANN significantly improved the classification accuracy.

Fourth, as discussed in Section V, we use techniques of adaptation operators and seed chromosome in our implementation of GA to improve the solution quality. To measure contributions of these two techniques, we conducted the ablation experiments, where we ran different versions of GA. Each version turns off certain features. For example, the adaptation operator consists of the mutation adaptation operator and the population adaptation operator. We, therefore, implemented three versions of GA, namely, GA_Mut_only (i.e., GA with the mutation adaptation operator only), GA_Pop_only (i.e., GA with the population adaptation operator only), and GA_No_Adp (i.e., GA without any adaptation operators). For the seed chromosome technique, we developed another version of GA, namely, GA_No_Seed (i.e., GA without seed chromosomes). Besides, GA_Full is the algorithm that we used in this study, which includes all the above techniques. Table VI shows experimental results on adaptation operators. If the adaptation operators were not used (see GA_No_Adp), the solution quality dropped sharply compared to GA_Full. If only a single adaptation operator was used, the solution quality also dropped and GA_Pop_only performed slightly better than GA_Mut_only. Table VII shows the experimental results on the seed chromosome. GA_Full (i.e., with seed chromosomes) performed substantially better than GA_No_Seed on train and test with the same set and performed slightly better on leave-one-out.

The ablation experiment results suggest that the two techniques, i.e., adaptation operators and seed chromosome, are both useful in improving the solution quality although the adaptation operators seem to play a more critical role.

TABLE VI ABLATION EXPERIMENTS ON ADAPTATION OPERATOR

Different Versions of GA	Train and Test with the Same Set	Leave-One-Out
GA_Full	82.81%	65.91%
GA_No_Adp	59.38%	47.73%
GA_Mut_only	56.25%	53.79%
GA_Pop_only	62.50%	58.33%

TABLE VII ABLATION EXPERIMENTS ON SEED CHROMOSOME

Different Versions of GA	Train and Test with the Same Set	Leave-One-Out
GA_Full	82.81%	65.91%
GA_No_Seed	70.31%	62.88%

Finally, we did not attempt to find the optimal ANN architecture that produces the best classification accuracy. Instead, we used the same architecture suggested by Weka [22] (when running its MultiLayerPerceptron function to perform ternary classification), which consists of three layers with 9 neurons being in the hidden layer as shown in Figure 5. All the experiments were conducted based on this architecture to ensure a fair comparison.

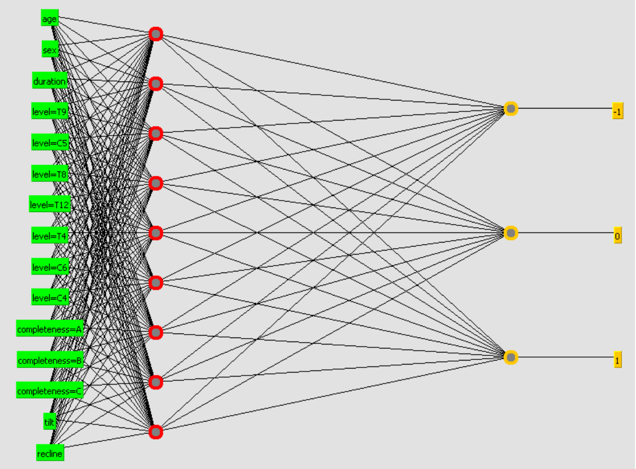

Figure 5. The Architecture of ANN

A. Study Limitations

First, this pilot study involved only 11 participants. We intended to investigate the feasibility of using ANN to classify/predict favorable wheelchair tilt and recline settings for people with SCI. We will recruit more participants to verify our results and refine the intelligent model. Second, the current intelligent model considers 5 attributes of the participants, namely, age, gender, duration of injury, level of injury, and completeness. In the subsequent study, we will consider more attributes that may affect the learning results. Finally, we investigated two thresholds, namely, 0.10 and 0.15, for ternary classification. In the subsequent study, we will investigate more thresholds and identify the one that produces the most meaningful results.

VIII. Conclusion

The purpose of this paper is to demonstrate the feasibility of using machine-learning techniques to classify/predict the favorable wheelchair tilt and recline settings for individual wheelchair users with SCI. We first present approaches that classify skin perfusion increase into two classes, namely, positive and negative. Then, we point out the major issue of binary classification, i.e., it may increase the chances of false positive and false negative classifications. To overcome this issue, we propose to perform ternary classification, i.e., classify skin perfusion increase into three classes, namely, positive, neutral, and negative. We investigated three approaches, namely, the traditional statistical approach, BP-based ANN, and GA-based ANN. Our experimental results show that the traditional approach that used the average skin perfusion to perform classifications could not accurately classify existing data. In comparison, using the BP algorithm to train ANN perfectly classified the existing data. However, the BP-based ANN could not generalize well. It performed poorly on the “leave-one-out” experiment, which was an 11-fold cross-validation. Hence, we used the genetic algorithm (GA) to train ANN in the “leave-one-out” experiment. Our experimental results show that the GA-based ANN significantly improved the generalization ability in comparison with the BP-based ANN and support vector machine (SVM) approaches. In summary, our experimental results provide suggestive evidence that using machine-learning techniques to classify favorable wheelchair tilt and recline settings is feasible.

In the next step, we will recruit more participants to verify and refine our ANN models. In addition, we will identify an optimal ANN architecture that will have good generalization ability to produce accurate prediction results.

References

[1] R. L. Johnson, K. A. Gerhart, J. McCray, J. C. Menconi, and G. G. Whiteneck, "Secondary conditions following spinal cord injury in a population-based sample," *Spinal cord,* vol. 36, pp. 45-50, Jan 1998.

[2] A. Gelis, A. Dupeyron, P. Legros, C. Benaim, J. Pelissier, and C. Fattal, "Pressure ulcer risk factors in persons with spinal cord injury part 2: the chronic stage," *Spinal Cord,* vol. 47, pp. 651-61, Sep 2009.

[3] N. S. C. I. S. Center, "Annual report for the Spinal Cord Injury Model Systems (public version)," Birmingham2006.

[4] M. J. Fuhrer, S. L. Garber, D. H. Rintala, R. Clearman, and K. A. Hart, "Pressure ulcers in community-resident persons with spinal cord injury: prevalence and risk factors," *Arch Phys Med Rehabil,* vol. 74, pp. 1172-7, Nov 1993.

[5] M. Reddy, S. S. Gill, and P. A. Rochon, "Preventing pressure ulcers: a systematic review," *JAMA : the journal of the American Medical Association,* vol. 296, pp. 974-84, Aug 23 2006.

[6] M. Lacoste, R. Weiss-Lambrou, M. Allard, and J. Dansereau, "Powered tilt/recline systems: why and how are they used?," *Assistive Technology,* vol. 15, pp. 58-68, 2003.

[7] S. M. Michael, D. Porter, and T. E. Pountney, "Tilted seat position for non-ambulant individuals with neurological and neuromuscular impairment: a systematic review.," *Clinical Rehabilitation,* vol. 21, pp. 1063-74, 2007.

[8] M. Reddy, S. S. Gill, and P. A. Rochon, "Preventing pressure ulcers: a systematic review," *JAMA,* vol. 296, pp. 974-84, 2006.

[9] M. Makhsous, M. Priebe, J. Bankard, D. Rowles, M. Zeigler, D. Chen, and F. Lin, "Measuring tissue perfusion during pressure relief maneuvers: insights into preventing pressure ulcers," *Journal of Spinal Cord Medicine,* vol. 30, pp. 497-507, 2007.

[10] J. Fu, Y.-K. Jan, and M. Jones, "Development of Intelligent Model to Determine Favorable Wheelchair Tilt and Recline Angles for People with Spinal Cord Injury," presented at the 33rd Annual International IEEE EMBS Conference Boston, pp. 2045-2048, MA, USA 2011.

[11] E. Alba and J. F. Chicano, "Training Neural Networks with GA Hybrid Algorithms," in *GECCO (1)* vol. 3102, ed: Springer, 2004, pp. 852-863.

[12] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature,* vol. 323, pp. 533-536, 1986.

[13] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*. Mass.: Addison-Wesley, 1989.

[14] B. L. Miller and D. E. Goldberg, "Genetic algorithms, selection schemes, and the varying effects of noise," *Evol. Comput.,* vol. 4, pp. 113-131, June 1996.

[15] F. G. Lobo and C. F. Lima, "Adaptive Population Sizing Schemes in Genetic Algorithms," in *Parameter Setting in Evolutionary Algorithms* vol. 54, ed: Springer, 2007, pp. 185-204.

[16] F. Rosenblatt, *Principles of Neurodynamics*. New York: Spartan Books, 1962.

[17] Y. K. Jan, M. A. Jones, M. H. Rabadi, R. D. Foreman, and A. Thiessen, "Effect of wheelchair tilt-in-space and recline angles on skin perfusion over the ischial tuberosity in people with spinal cord injury," *Archives of physical medicine and rehabilitation,* vol. 91, pp. 1758-64, Nov 2010.

[18] M. Makhsous, M. Priebe, J. Bankard, D. Rowles, M. Zeigler, D. Chen, and F. Lin, "Measuring tissue perfusion during pressure relief maneuvers: insights into preventing pressure ulcers," *The journal of spinal cord medicine,* vol. 30, pp. 497-507, 2007.

[19] S. L. Garber, D. H. Rintala, K. A. Hart, and M. J. Fuhrer, "Pressure ulcer risk in spinal cord injury: predictors of ulcer status over 3 years," *Arch Phys Med Rehabil,* vol. 81, pp. 465-71, Apr 2000.

[20] G. J. McLachlan, K. A. Do, and C. Ambroise, *Analyzing microarray gene expression data*: Wiley, 2004.

[21] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery,* vol. 2, pp. 121-167, 1998.

[22] I. Witten and E. Frank, *Data mining: Practical machine learning tools and techniques*. San Francisco: Morgan Kaufmann, 2005.

Acknowledgment

This work was supported by the National Institutes of Health (NIH) (P20RR016478) through Oklahoma IDeA Network of Biomedical Research Excellence (INBRE) -- Junior Investigator Award and the National Institutes of Health (NIH) (R03HD060751).