

Automated Activity Recognition with Gait Positions Using Machine Learning Algorithms

Yeuk Lam Ng

Faculty of Arts & Science,
University of Toronto,
Canada

Xilin Jiang

Grainger College of Engineering,
University of Illinois at Urbana-
Champaign, USA

Yinghui Zhang

College of Letters and Science,
University of California, Berkeley, USA

Seung Bin Shin

Hankuk Academy of Foreign Studies,
Korea

Ruoci Ning

Austin Preparatory School,
USA

Abstract—Exoskeletons are wearable devices for enhancing human physical performance and for studying actions and movements. They are worn on the body for additional power and load-carrying capacity. Exoskeletons can be controlled using signals from the muscles. In recent years, gait analysis has attracted increasing attention from fields such as animation, athletic performance analysis, and robotics. Gait patterns are unique, and each individual has his or her own distinct gait pattern characteristics. Gait analysis can monitor activity in sensitive areas. This paper uses various machine learning algorithms to predict the activity of subjects using exoskeletons. Here, localization data from the UIC machine learning repository are used to recognize activities with gait positions. The study also compares five machine learning methods and examines their efficiency and accuracy in activity prediction for three different subjects. The results for the various machine learning methods along with efficiency and accuracy results are discussed.

Keywords—exoskeletons; gait analysis; activity recognition; machine learning algorithms

I. INTRODUCTION

Exoskeletons are wearable devices for enhancing human performance. They can supplement or increase the wearer's speed, strength, and endurance. The wearer applies some force that acts to send control signals to the robotic system. This signal is sensed and amplified by the exoskeleton to exert the requisite force for the desired movement [1]. Exoskeletons have gained increasing attention in recent years when the U.S. Defense department allotted funds for exoskeletons for human performance augmentation. There is a wide range of applications for exoskeletons, including walking assistance for physically impaired people.

A. Importance of Wearable Exoskeletons

One important application of exoskeletons is walking assistance [2, 4]. Walking or movement assistance is achieved when exoskeletons are used for the augmentation of a healthy individual's physical ability such as running or in the rehabilitation of physically disabled individuals. In rough

terrain or staircases, moving heavy things is difficult using wheeled vehicles. The use of automated or semi-automated human-like legs can be an alternative option since legs can adapt to a wide range of extreme surfaces. The Berkeley Lower Extremity Exoskeleton (BLEEX) allows its wearer to carry large loads on his or her back with minimal effort over various kinds of terrain [3]. The exoskeleton market's worth was about \$200 million in 2017 [5], and is estimated to reach \$1.3 billion by 2024 [6]. This necessitates reliable and accurate methods for predicting human movement.

B. Movement Prediction in Exoskeleton Robots

The synchronization of the exoskeleton system with the wearer in motion and the application of force are of intense research interest. The ultimate objective is to actively support the wearer's desired movement. The inherent periodicity of movement hampers oscillator-based approaches for modeling and motion. An oscillator model is used for coordinated periodic trajectories of exoskeletons, where the wearer's movement signals act as references to the robotic controller [6]. Therefore, accuracy in predicting human motion from exoskeleton movement is crucial for estimating exoskeleton and human motion using sensors for effective human-robot interface. By being independent of any external measurement, this approach can provide precise measurement in real-world situations [7]. A precursor to proper exoskeleton operation is the ability to correctly estimate the wearer's intended movement such as walking and running. The ability of the exoskeleton to recognize intended movement is crucial since controlling muscles and joints in walking is different from that in running or climbing. Therefore, the device should predict the operator's trajectory. Finally, it is crucial to generate correct actuator signals for intended trajectories in real time [8].

II. LITERATURE REVIEW

Authors in [9] proposed a control scheme based on machine movement synchronization where a neural oscillator model is used for motion assistance. Authors in [10] developed an oscillator-based motion assistance protocol for medical therapy

Corresponding author: Y. L. Ng (jingwensun321@gmail.com)

for exoskeleton use. The protocol was experimentally evaluated for a gait rehabilitation robot [11]. In [12-14], authors considered the adaptation of a mechatronic device to human movement. They segregated human gait patterns into spatial and temporal. The former represents personal movement style, whereas the latter is for any movement cycle. Style and phase were separated from observed walking behaviors [12] to control exoskeletons [13-14]. The proposed style-phase separation approach was adapted to multi-user movement based on rapid online calibration mechanisms for predicting the wearer's movement [6].

In [15], authors developed a method for obtaining joint angles, angular velocity, and force/torque from exoskeleton sensors and used them to distinguish different movements including standing, walking and sitting. In [16], they developed a model for predicting joint trajectories of the wearer and in [17] they evaluated leg dynamics for gait trajectory references for any movements. Authors in [8] considered a class of unsupervised machine learning algorithms for intended movements and future trajectories for a lower limb exoskeleton. They considered the conditional restricted Boltzmann machine, which can predict intended movements and trajectories on a real-time basis. Authors in [18] used a multi-layer perceptron neural network to predict natural gait parameters and used age, gender, height, and weight, among others, as training input. The output included desired walking speed and cadence. Two neural networks were trained to analyze efficiency and accuracy in the prediction of gait patterns involving slow speeds and then both slow and normal speeds [18]. In [19], three supervised machine learning techniques for predicting activation patterns for muscles and sensory data from time series for human movement for an exoskeleton for assisted walking were presented. Authors considered a multilayer perceptron with modified back-propagation learning algorithm, an adaptive neuro-fuzzy inference system, and an entropy-based inductive learning model. Training and testing data were used for the assisted controller. The minimal and most comprehensible rules were obtained through the adaptive neuro-fuzzy inference system. The IL-RBF network provided the best generalization. In [20], authors used Elman's recurrent neural network (ERNN) to automatically identify healthy and pathological gait. They considered stance, swing and double support intervals and developed two schemes to classify healthy individuals from patients with some disease, reporting an average performance value of 85%.

III. MACHINE LEARNING ALGORITHMS

Machine learning employs artificial intelligence with data mining to learn from experience without the need for explicit programming. This learning is performed either directly or by observing patterns, and its main objective is to make computers learn on their own without human intervention. The two most common forms of machine learning are supervised and unsupervised, followed by semi-supervised learning and reinforcement learning. Supervised learning employs training using positive and negative examples, where the algorithm receives inputs and class labels as positive or negative. Some examples of supervised learning include classification,

regression, recommendation, and prediction systems. This type of learning anticipates future occurrences using previous (historical) data. Unsupervised learning learns through historical data but with no historical labels. There is no class label for determining positive or negative examples. Clusters or segments are formed with similarities in existing data. Popular unsupervised learning methods include k-means clustering, nearest neighbor, and self-organizing maps. Semi-supervised learning employs labeled or unlabeled historical data and it is used with supervised learning methods including classification, regression, and prediction. This type of learning is useful when it is difficult or costly to obtain class labels for all given data. In this case, some datasets contain class labels, while others do not. The final category of learning is reinforcement learning. Here, the algorithm learns by trial and error. Reinforcement learning is the most common in AI problems. Its algorithm has three main components: agents, environment and actions.

In exoskeletons, material heterogeneity, morphological incongruities, or dynamically asynchronous movements cause some relative motion between the exoskeleton and the wearer's body. This must be minimized for efficient power transmission, effective limb control, and low pain from movement [7]. Accurate and rapid prediction of gait patterns enables synchronous movement by reducing this human-machine relative motion.

IV. METHODOLOGY

This study employs both supervised and unsupervised machine learning algorithms to classify human activity using a dataset of activity recordings using 4 tags from the UCI Machine Learning repository. The experiments were conducted using WEKA with various algorithms. Accordingly, 7 different machine learning algorithms were used in this study, namely Naive Bayes, KNN, support vector machine, J48 decision tree, radial basis function network classifier, multi-layer perceptron neural network, and random forest. The Localization Data for Person Activity dataset were downloaded from the UCI Machine Learning Repository [21]. The dataset contained 164,860 instances with 8 attributes for each instance. The dataset was univariate, sequential and in time series. This dataset was suitable for classification tasks in machine learning and provided recordings for 5 individuals who wore 4 tags at left ankle, right ankle, belt and chest. Each instance was a localization data point for one tag. The tag attribute has 4 different labels with corresponding equivalent value in the dataset. Localization data for tags were collected at 3 coordinates, x, y, and z. The class label for the dataset was activity containing 11 types of activities. The dataset contained 2 additional attributes: time stamp and date. Since the study did not consider time series activities, these 2 attributes were removed in the data preprocessing stage. The resulting data were stored in the database in WEKA data format.

The experiments involved 3 individuals for the performance of various machine learning classification algorithms, so the data were further processed to separate the datasets for each person's activity. Then the localization datasets for these 3 individuals were loaded to WEKA to build various machine learning models by passing through machine learning algorithms. The models were tested using 10-fold cross

validation testing for a better performance evaluation. The experimental setup was run for 3 individuals and 7 algorithms.

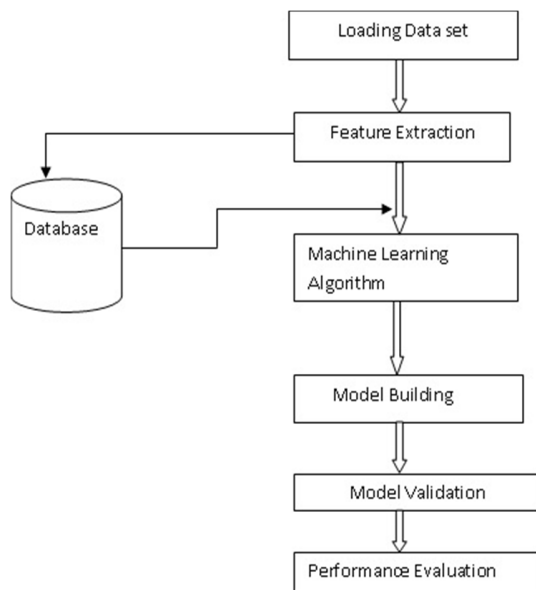


Fig. 1. Process flow diagram

V. EXPERIMENTS

Gait pattern prediction is an important challenge in robotic gait rehabilitation. The key parameters are gait pattern, walking speed, cadence and stride length and require careful analysis. Gait parameters in natural walking must be accurately predicted, and accurate gait pattern recognition can be implemented using various machine learning algorithms and proper datasets. We conducted experiments on 7 popular data mining algorithms and the results are tabulated for each algorithm. The results are analyzed and the performances are compared below. The following classes are used for the class attributes in the used dataset: a=walking, b=sitting down, c=sitting, d=standing up from sitting, e=falling, f=lying, g=standing up from lying, h=lying down, i=sitting on the ground, j=standing up from sitting on the ground, k=on all fours. Considering the above class labels, experiments were conducted with 7 different data mining algorithms. For each algorithm the TP Rate, FP Rate, Precision, Recall, F-Measure, ROC curve are tabulated for all class labels. Also, for each algorithm the number of correctly classified instances, incorrectly classified instances, Kappa statistic and 3 different error measures, namely root mean squared error, relative absolute error and root relative squared error are tabulated. The detailed accuracies by class variable and the run information of the seven data mining algorithms are presented in Tables I to VII and the final comparison of the 7 algorithms is exhibited in Table VIII.

TABLE I. NAIVE BAYES ALGORITHM

Class	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Summary		
A	0.299	0.032	0.585	0.299	0.396	0.360	0.855	0.512	Correctly Classified Instances	12804	46.6058%
B	0.000	0.000	?	0.000	?	?	0.819	0.037	Incorrectly Classified Instances	14669	53.3942%
C	0.493	0.136	0.449	0.493	0.470	0.344	0.834	0.556	Kappa statistic	0.2784	
D	0.000	0.000	?	0.000	?	?	0.853	0.019	Mean absolute error	0.1181	
E	0.000	0.000	0.000	0.000	0.000	0.001	0.838	0.145	Root mean squared error	0.2428	
F	0.694	0.394	0.525	0.694	0.598	0.292	0.772	0.689	Relative absolute error	83.2151%	
G	0.014	0.004	0.257	0.014	0.027	0.041	0.607	0.140	Root relative squared error	91.1429%	
H	0.002	0.002	0.037	0.002	0.005	0.002	0.818	0.082	Total number of instances	27473	
I	0.597	0.137	0.291	0.597	0.391	0.338	0.842	0.249			
J	0.000	0.000	?	0.000	?	?	0.870	0.095			
K	0.355	0.023	0.413	0.355	0.382	0.357	0.951	0.400			
L	0.466	0.195	?	0.466	?	?	0.798	0.494			

TABLE II. K-NEAREST NEIGHBOUR

Class	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Summary		
A	0.720	0.040	0.736	0.720	0.728	0.687	0.840	0.577	Correctly Classified Instances	21799	79.347%
B	0.221	0.007	0.237	0.221	0.228	0.221	0.612	0.061	Incorrectly Classified Instances	5674	20.653%
C	0.898	0.029	0.874	0.898	0.886	0.860	0.935	0.811	Kappa statistic	0.7347	
D	0.272	0.004	0.276	0.272	0.274	0.270	0.634	0.079	Mean absolute error	0.0376	
E	0.253	0.010	0.298	0.253	0.274	0.264	0.625	0.091	Root mean squared error	0.1937	
F	0.906	0.068	0.893	0.906	0.900	0.836	0.920	0.852	Relative absolute error	26.4912%	
G	0.506	0.044	0.532	0.506	0.518	0.472	0.728	0.323	Root relative squared error	72.7205%	
H	0.296	0.019	0.323	0.296	0.309	0.289	0.634	0.119	Total number of instances	27473	
I	0.949	0.007	0.931	0.949	0.940	0.934	0.972	0.890			
J	0.576	0.007	0.582	0.576	0.579	0.572	0.778	0.374			
K	0.679	0.015	0.669	0.679	0.674	0.659	0.834	0.478			
L	0.793	0.043	0.788	0.793	0.791	0.750	0.875	0.693			

TABLE III. SUPPORT VECTOR MACHINE

Class	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Summary		
A	0.474	0.034	0.678	0.474	0.558	0.514	0.720	0.391	Correctly Classified Instances	18662	67.9285%
B	0.000	0.000	?	0.000	?	?	0.500	0.010	Incorrectly Classified Instances	8811	32.0715%
C	0.749	0.075	0.692	0.749	0.719	0.654	0.837	0.564	Kappa statistic	0.5623	
D	0.000	0.000	?	0.000	?	?	0.500	0.550	Mean absolute error	0.0583	
E	0.002	0.000	1.000	0.002	0.004	0.047	0.501	0.018	Root mean squared error	0.2415	
F	0.937	0.280	0.678	0.937	0.786	0.641	0.828	0.659	Relative absolute error	41.0753%	
G	0.102	0.008	0.569	0.102	0.173	0.215	0.547	0.140	Root relative squared error	90.6403%	
H	0.002	0.000	0.222	0.002	0.005	0.021	0.501	0.030	Total number of instances	27473	
I	0.937	0.025	0.770	0.901	0.830	0.816	0.938	0.702			
J	0.006	0.000	0.300	0.006	0.012	0.042	0.503	0.019			
K	0.702	0.028	0.528	0.702	0.603	0.589	0.837	0.384			
L	0.679	0.130	?	0.679	?	?	0.774	0.501			

TABLE IV. RBF CLASSIFIER

Class	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Summary		
A	0.041	0.003	0.683	0.041	0.077	0.145	0.669	0.260	Correctly Classified Instances	12521	45.5757%
B	0.000	0.000	?	0.000	?	?	0.493	0.010	Incorrectly Classified Instances	14952	54.4243%
C	0.638	0.260	0.356	0.638	0.457	0.311	0.784	0.503	Kappa statistic	0.2032	
D	0.000	0.000	?	0.000	?	?	0.495	0.005	Mean absolute error	0.1303	
E	0.000	0.000	?	0.000	?	?	0.513	0.022	Root mean squared error	0.2515	
F	0.863	0.536	0.503	0.863	0.635	0.336	0.798	0.704	Relative absolute error	91.7506%	
G	0.000	0.000	?	0.000	?	?	0.579	0.112	Root relative squared error	94.4148%	
H	0.000	0.000	?	0.000	?	?	0.591	0.045	Total number of instances	27473	
I	0.000	0.000	?	0.000	?	?	0.709	0.200			
J	0.000	0.000	?	0.000	?	?	0.554	0.022			
K	0.008	0.000	0.015	0.008	0.015	0.06	0.641	0.097			
L	0.456	0.255	?	0.456	?	?	0.724	0.432			

TABLE V. MULTI LAYER PERCEPTRON

Class	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Summary		
A	0.563	0.062	0.583	0.563	0.573	0.509	0.892	0.591	Correctly Classified Instances	19084	69.4646%
B	0.000	0.000	0.000	0.000	0.000	-0.002	0.789	0.029	Incorrectly Classified Instances	8389	30.5354%
C	0.792	0.085	0.677	0.792	0.730	0.666	0.908	0.768	Kappa statistic	0.5965	
D	0.000	0.000	0.000	0.000	0.000	-0.001	0.834	0.024	Mean absolute error	0.076	
E	0.117	0.003	0.394	0.117	0.180	0.208	0.786	0.138	Root mean squared error	0.2012	
F	0.899	0.160	0.780	0.899	0.835	0.725	0.929	0.872	Relative absolute error	53.5013%	
G	0.220	0.029	0.435	0.220	0.292	0.263	0.704	0.282	Root relative squared error	75.5171%	
H	0.017	0.001	0.500	0.017	0.033	0.089	0.834	0.134	Total number of instances	27473	
I	0.882	0.018	0.822	0.882	0.851	0.837	0.962	0.843			
J	0.072	0.004	0.264	0.072	0.113	0.130	0.868	0.159			
K	0.644	0.036	0.443	0.644	0.525	0.509	0.942	0.447			
L	0.695	0.091	0.658	0.695	0.661	0.596	0.895	0.683			

TABLE VI. J48 DECISION TREE

Class	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Summary		
A	0.709	0.049	0.689	0.709	0.699	0.652	0.877	0.631	Correctly Classified Instances	21517	78.3205
B	0.118	0.005	0.195	0.118	0.147	0.145	0.724	0.089	Incorrectly Classified Instances	5956	21.6795
C	0.895	0.036	0.850	0.895	0.872	0.842	0.953	0.848	Kappa statistic	0.7182	
D	0.143	0.002	0.280	0.143	0.189	0.197	0.730	0.098	Mean absolute error	0.048	
E	0.157	0.007	0.267	0.157	0.198	0.195	0.702	0.112	Root mean squared error	0.1818	
F	0.929	0.091	0.865	0.929	0.896	0.828	0.942	0.868	Relative absolute error	33.7819	
G	0.428	0.038	0.527	0.428	0.472	0.428	0.777	0.382	Root relative squared error	68.2489	
H	0.206	0.013	0.330	0.206	0.254	0.243	0.772	0.179	Total number of instances	27473	
I	0.930	0.008	0.918	0.930	0.924	0.917	0.973	0.880			
J	0.436	0.006	0.541	0.436	0.483	0.478	0.812	0.329			
K	0.677	0.016	0.657	0.677	0.667	0.652	0.908	0.570			
L	0.783	0.054	0.763	0.783	0.771	0.728	0.907	0.723			

TABLE VII. RANDOM FOREST

Class	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Summary		
									Correctly Classified Instances	25241	79.8943%
A	0.816	0.063	0.772	0.816	0.793	0.738	0.964	0.882	Incorrectly Classified Instances	6352	20.1057%
B	0.202	0.003	0.401	0.202	0.268	0.279	0.881	0.227	Kappa statistic	0.7482	
C	0.917	0.030	0.862	0.917	0.889	0.866	0.989	0.954	Mean absolute error	0.0506	
D	0.366	0.003	0.556	0.366	0.442	0.447	0.949	0.389	Root mean squared error	0.1607	
E	0.218	0.004	0.444	0.218	0.293	0.304	0.908	0.296	Relative absolute error	34.4957%	
F	0.933	0.063	0.873	0.933	0.902	0.855	0.983	0.960	Root relative squared error	59.3177%	
G	0.557	0.042	0.643	0.557	0.597	0.548	0.916	0.652	Total number of instances	31593	
H	0.414	0.013	0.574	0.414	0.482	0.470	0.934	0.505			
I	0.949	0.006	0.911	0.949	0.930	0.925	0.998	0.972			
J	0.529	0.006	0.628	0.529	0.575	0.569	0.965	0.622			
K	0.581	0.012	0.585	0.581	0.583	0.571	0.976	0.617			
L	0.799	0.044	0.785	0.799	0.789	0.753	0.968	0.848			

TABLE VIII. COMPARISON OF THE 7 MACHINE LEARNING ALGORITHMS

Algorithm	Experiment – I		Experiment – II		Experiment – III	
	RMS Error	Performance	RMS Error	Performance	RMS Error	Performance
Naive Bayes	0.2428	46.60%	0.2483	46.64%	0.2471	46.60%
K-Nearest Neighbor	0.1937	79.34%	0.1933	79.43%	0.2012	77.74%
SVM	0.2415	67.92%	0.2568	63.71%	0.2645	61.51%
RBF Classifier	0.2515	45.57%	0.2581	42.82%	0.2601	40.56%
Multi Layer Perceptron	0.2012	69.46%	0.2103	65.62%	0.2144	64.06%
J48 Classifier	0.1818	78.32%	0.1848	77.59%	0.1902	76.24%
Random Forest	0.1536	81.82%	0.1556	81.43%	0.1607	79.89%

A. Naive Bayes

In machine learning models, the classifiers are used to discriminate different objects based on certain features in the datasets. Naive Bayes classifier is a probabilistic machine learning model used for classification problems using the Bayes theorem. In that way, we can find the probability of event A happening, given that B has occurred. Here, B is the evidence and A is the hypothesis. The assumption made here is that the predictors/features are independent. The presence of one particular feature does not affect the other. Hence it is called naive.

B. Lazy IBK

Lazy learning algorithms in machine learning try to generalize the training data and the results are delayed until a query is made to the system, as opposed to eager learning, where the system tries to generalize the training data before receiving queries. Generally, for employing lazy learning, K-nearest neighbor algorithm is implemented. The advantage in employing a lazy learning method is that the target function will be approximated locally as in the k-nearest neighbor algorithm. The disadvantages with lazy learning include the large space requirement to store the complete training dataset. Mostly noisy training data increase the case support unnecessarily, because no concept is made during the training phase. Another disadvantage is that lazy learning methods are usually slower to evaluate, though this is accompanied with a faster training phase.

C. SVM

A Support Vector Machine (SVM) performs classification by finding the hyperplane that maximizes the margin between two classes. SVMs are perhaps one of the most popular and talked about machine learning algorithms. They were

extremely popular around the time they were developed in the '90s and continue to be the go-to method for high-performing algorithms with little tuning. The learning of the hyperplane in linear SVM is done by transforming the problem using linear algebra.

D. Multi Layer Perceptron

A multilayer perceptron (MLP) is a class of feed forward artificial neural networks (ANNs). An MLP consists of at least three layers of nodes. Except for the input nodes, each node is a neuron that uses a nonlinear activation function. MLP utilizes a supervised learning technique called back propagation for training. An ANN has several advantages, but one of the most recognized ones is the fact that it can actually learn from observing data sets. In this way, the ANN is used as a random function approximation tool. These types of tools help estimate the most cost-effective methods for arriving at solutions while defining computing functions or distributions. ANN takes data samples rather than entire data sets to arrive at solutions, which saves both time and money. ANNs are considered fairly simple mathematical models to enhance existing data analysis technologies.

E. Random Forest

Random Forest is one of the most popular and most powerful machine learning algorithms. It is a type of ensemble machine learning algorithm called Bootstrap Aggregation or bagging. Ensembled algorithms are those which combine more than one algorithms of the same or different kind for classifying objects. Random Forest classifier creates a set of decision trees from randomly selected subsets of the training set. It then aggregates the votes from different decision trees to decide the final class of the test object.

F. J48

J48 is the java implementation of C4.5 which is a successor of ID3. It is used for classification in which new data are labeled according to already existing observations (training data set). Decision tree induction begins with a dataset (training set) which is partitioned at every node resulting in smaller partitions, thus following a recursive divide and conquer strategy. In addition to a data set, which is a collection of objects, a set of attributes is also passed. Objects can be an event or an activity and the attributes are the information related to that object. Every tuple in the data set is associated with a class label which identifies whether an object belongs to a particular class or not. Splitting can further be performed only if the tuples fall in different classes.

G. RBF Classifier

A Radial Basis Function Network (RBFN) is a particular type of neural network. RBFN performs classification by measuring the input's similarity to examples from the training set. Each RBFN neuron stores a "prototype", which is just one of the examples from the training set. When we want to classify a new input, each neuron computes the Euclidean distance between the input and its prototype. It consists of an input vector, a layer of RBF neurons, and an output layer with one node per category or class of data. Each RBF neuron stores a "prototype" vector which is just one of the vectors from the training set.

VI. RESULTS AND DISCUSSION

The study used the Localization Data for Person Activity dataset from the UCI Machine Learning Repository. The dataset contained 164,860 instances with 8 attributes. The dataset consisted of recorded data by 5 people who wore 4 tags (exoskeletons) on left ankle, right ankle, belt and chest. Each instance was a localization data point for one exoskeleton. Each data instance contained sequence number, tag identifier, timestamp, date format, x-coordinate, y-coordinate, z-coordinate, and activity. Activity values were walking, falling, lying down, lying, sitting down, sitting, standing up from lying, on all fours, sitting on the ground, standing up from sitting, standing up from sitting on the ground. A total of 7 machine learning models were built using 3 different datasets. Tables I to VII provide the detailed accuracy and run information of 7 ML algorithms and Table VIII provides their comparison. According to the results the k-nearest neighbor algorithm outperformed RBF and multi-layer perceptron, but Random Forest outperformed all other algorithms in all three experiments.

VII. CONCLUSION

This study compares 7 machine learning algorithms using WEKA for the classification of action using a dataset on a person's exoskeleton activity. The results show that the Random Forest algorithm provided good classification performance, whereas the radial basis function and Naive Bayes provided poor performance. Further, the results of three experiments for different datasets show that the Random Forest algorithm outperformed the multilayer perceptron neural network in activity recognition.

ACKNOWLEDGEMENT

The authors acknowledge the assistance of Seokwoo Chung (Shanghai American School) in literature review.

REFERENCES

- [1] I. S. Dhindsa, R. Agarwal, H. S. Ryaat, "A novel algorithm to predict knee angle from EMG signals for controlling a lower limb exoskeleton", Information Technology and Nanotechnology, Samara, Russia, May, 2016
- [2] J. O. Brinker, T. Matsubara, T. Teramae, T. Noda, T. Ogasawarsa, T. Afour, J. Morimoto, "Walking Pattern Prediction with Partial Observation for Partial Walking Assistance by using an Exoskeleton System", IEEE International Conference on Rehabilitation Robotics, Singapore, August 11-14, 2015
- [3] A. Zoss, H. Kazerooni, A. Chu, "On the Mechanical Design of the Berkeley Lower Extremity Exoskeleton (BLEEX)", IEEE/RSJ International Conference on Intelligent Robots and Systems, Edmonton, Canada, August 2-6, 2005
- [4] A. M. Dollar, H. Herr, "Lower Extremity Exoskeletons and Active Orthoses: Challenges and State-of-the-Art", IEEE Transactions on Robotics, Vol. 24, No. 1, pp. 144-158, 2008
- [5] M. J. Coren, "Robot exoskeletons are finally here, and they're nothing like the suits from Iron Man", available at: <https://qz.com/971741/robot-exoskeletons-are-finally-here-and-theyre-nothing-like-the-suits-from-iron-man/>, 2017 (accessed 27 June 2018)
- [6] V. Verma, A. Agarwal, "Exoskeleton Market Size, By Product (Stationary, Mobile), Type (Full Body, Upper Body, Lower Body), By Application (Industrial, Military, Healthcare) Industry Analysis Report, Regional Outlook (U.S., Canada, Germany, France, UK, Russia, China, Japan, South Korea, Brazil, Mexico, MEA), Application Development, Competitive Landscape & Forecast, 2017 - 2024", available at: <https://www.gminsights.com/industry-analysis/exoskeleton-market>, 2017 (accessed 27 June 2018).
- [7] D. Torricelli, C. Cortes, N. Lete, A. Bertelsen, J. E. Gonzalez-Vargas, A. J. Del-Ama, I. Dimbwadyo, J. C. Moreno, J. Florez, J. L. Pons, "A subject-specific kinematic model to predict human motion in exoskeleton-assisted gait", Frontiers in Neurobotics, Vol. 12, 2018
- [8] E. Chong, F. C. Park, "Movement prediction for a lower limb exoskeleton using a conditional restricted Boltzmann machine", Robotica, Vol. 35, No. 11 pp. 2177-2200, 2017
- [9] X. Zhang, M. Hashimoto, "SBC for motion assist using neural oscillator", IEEE International Conference on Robotics and Automation, Kobe, Japan, May 12-17, 2009
- [10] R. Ronsse, N. Vitiello, T. Lenzi, J. van den Kieboom, M. C. Carrozza, A. J. Ijspeert, "Human-robot synchrony: Flexible assistance using adaptive oscillators", IEEE Transactions on Biomedical Engineering, Vol. 58, No. 4, pp. 1001-1012, 2011
- [11] R. Ronsse, B. Koopman, N. Vitiello, T. Lenzi, D. Rossi, S. M. Maria, J. van den Kieboom, E. van Asseldonk, M. C. Carrozza, H. van der Kooij, A. J. Ijspeert, "Oscillator-based Walking Assistance: a Model-free Approach", IEEE International Conference on Rehabilitation Robotics, Zurich, Switzerland, June 29-July 1, 2011
- [12] T. Matsubara, S. H. Hyon, J. Morimoto, "Real-time stylistic prediction for whole-body human motions", Neural Networks, Vol. 25, pp. 191-199, 2012
- [13] T. Matsubara, A. Uchikata, J. Morimoto, "Full-body exoskeleton robot control for walking assistance by style-phase adaptive pattern generation", IEEE/RSJ International Conference on Intelligent Robots and Systems, Vilamoura, Portugal, October 7-12, 2012
- [14] T. Matsubara, D. Uto, T. Noda, T. Teramae, J. Morimoto, "Style-phase adaptation of human and humanoid biped walking patterns in real systems", IEEE-RAS International Conference on Humanoid Robots (Humanoids), Madrid, Spain, November 18-20, 2014
- [15] H. A. Varol, F. Sup, M. Goldfarb, "Multiclass real-time intent recognition of a powered lower limb prosthesis", IEEE Transactions on Biomedical Engineering, Vol. 57, No. 3, pp. 542-551, 2010

- [16] L. Wang, E. H. Van Asseldonk, H. Van Der Kooij, "Model Predictive Control-Based Gait Pattern Generation for Wearable Exoskeletons", IEEE International Conference on Rehabilitation Robotics, Singapore, August 11-14, 2011
- [17] J. F. Veneman, R. Kruidhof, E. E. Hekman, R. Ekkelenkamp, E. H. van Asseldonk, H. van der Kooij, "Design and evaluation of the LOPES exoskeleton robot for interactive gait rehabilitation", IEEE Transactions on Neural Systems and Rehabilitation Engineering, Vol. 15, No. 3, pp. 379-386, 2007
- [18] H. B. Lim, T. P. Luu, K. H. Hoon, K. H. Low, "Natural gait parameters prediction for gait rehabilitation via artificial neural network IEEE/RSJ International Conference on Intelligent Robots and Systems, Taipei, Taiwan, October 18-22, 2010
- [19] S. Jonic, T. Jankovic, V. Gajic, D. Popovic, "Three machine learning techniques for automatic determination of rules to control locomotion", IEEE Transactions on Biomedical Engineering, Vol. 46, No. 3, pp. 300-310, 1999
- [20] S. Dutta, A. Chatterjee, S. Munshi, "An automated hierarchical gait pattern identification tool employing cross - correlation - based feature extraction and recurrent neural network based classification", Expert Systems, Vol. 26, No. 2, pp. 202-217, 2009
- [21] <https://archive.ics.uci.edu/ml/datasets/Localization+Data+for+Person+Activity>