OPTIMAL FEATURE SELECTION FOR POWER AWARE COMPUTING IN WEARABLE SENSOR NETWORKS

[1]Nithyapriya G,[2]Prof.Kumaresan S
[1]PG Student,[2]Professor and Head of the Department
Department of Computer Science and Engineering
Government college of Technology,Coimbatore,India.
[1]nithyapriya94@gmail.com

Abstract-***Recent technology advances have led to the development of different sensing, computing and communication capabilities that are becoming essential in our daily lives. A special case of this platform is wearable sensor networks whose computational elements are coupled with the body. There are a number of challenges in implementing this including high cost, less weight, power efficiency and battery life. Since they are battery operated and used for life-saving purposes, power efficiency is considered as most challenging. Previous studied revealed that the sensing, processing and data transmissions incur high energy expenditure. On-node processing was introduced which employed signal processing and machine learning techniques. Current technologies uses a vast set of features that are extracted from the sensor data for a specific application such as activity recognition where more power is consumed for data processing. Hence, this work focuses on reduced set of features that are optimal based on redundancy and relevance analysis. These features are determined prior to signal processing which reduces the processing time and hence the power consumption. It takes into consideration the energy cost of individual features that are calculated in real-time. A graph model is introduced to represent correlation and computing complexity of the features. The problem is formulated using integer programming and a greedy approximation ispresented to select the features in a power-efficient manner.***

Key Terms* – *Real-time systems and embedded systems, low-power design, wearable computers, healthcare, optimization.

I. INTRODUCTION

Wearable sensory devices are becoming the enabling technology for many applications in healthcare and well-being, where computational elements are tightly coupled with the human body to monitor specific events about their subjects.Theseubiquitous systems have proved to be effective in a numberof domains ranging from medical [1] and well-being [2] tomilitary and smart vehicles . A special class of theseplatforms is wearable sensor networks whose computationalelements are tightly coupled with the human body.These networks are known as enabling technologies formany applications such as remote patient monitoring and personalized healthcare , gaming and sports , maintenance production and process support. There are a number of challenges that must be overcometo fully implement wearable sensor networks includinghigh costs, package size and weight limitations, power efficiencyand battery lifetime, memory storage, connectivity,ease of use, reliability, application level accuracy, security, and privacy issues . Since wearable sensor networks arebattery-operated and may have critical and life-saving purposes,power efficiency is considered the most challengingdesign consideration in their real life deployment. In medicalapplications, wearable units are mainly used for remoteand continuous patient monitoring , and therefore, theirpower consumption needs to be minimized to guaranteetheir long term operation and infrequent battery charge or replacement.

In wearable sensor networks, where raw data is simply streamed to the gateway, the largest energy consumer is the radio subsystem, with the processing unit only required for formatting the data according to the utilized communications protocol. On the other hand, for wearable systems with on-node processing (e.g., movement monitoring and wearable EEG monitors), the processing subsystem is the most energy consuming subsystem. In such systems, a signal with a lower bit rate will be transmitted to the gateway after processing. This necessitates further optimization of the computing units' power consumption in order to prolong the lifetime of the entire system. This second group of wearable systems often employ embedded signal processing and machine learning blocks that use sensor data (e.g., acceleration of body segments) to extract relevant information (e.g., types of movements about their subjects..

Although classification accuracy is the ultimate measure of relative performance, for wearable platforms, there should be a mechanism to gauge the amount of useful information extracted for a given energy budget. Hence, real-life deployment of wearable platforms necessitates the incorporation of energy components into performance measures. The traditional feature selection algorithms focus on specific criterion that finds redundancy and relevance in a given feature set. This approach is generally acceptable in conventional algorithms such as image processing and text mining, which run on highly powerful computers. These techniques, however, do not take into consideration computing complexity of individual features. That is, they allocate equal weights to features of varying complexity. In wearable systems this approach is not effective, as these systems have limited processing power and need to operate in real-time. None of the feature selection techniques studied in the past takes into consideration the computing complexity of the selected features, an important measure in designing wearable monitoring systems.

Furthermore, none of the existing power-awareschemes in embedded system design has dealt with featureselection algorithms and how the energy saving mechanismscan cleverly prevent some features from being accounted for in classification mechanisms.This study embodies the innovation of the notion ofpower-aware feature selection; we model the problem ofenergy optimal feature set and prove that it constitutes acomputational problem that is NP-hard and finally we providean approximation to find the appropriate minimumcost feature set. Real human motion data sets are used inorder to verify the efficacy of our approximation scheme.

II. RELATED WORK

In the context of real-time computing, energy-reliability tradeoff was explored[3]. Their approach minimizes the system-level energy consumption while satisfying a certain reliability target in the task scheduler. More specifically, their approach specifies the optimal number of recoveries to deploy together with task-level processing frequencies to minimize the energy consumption while achieving the target reliability and meeting the deadline constraints.

A new method called dynamic sensor selection was proposed in [4], which trades off classification accuracy for battery lifetime in wireless sensor networks. Dynamic sensor selection investigates the impact of varying number of sensors on activity classification accuracy and minimizes the number of nodes necessary to obtain a given classification accuracy for activity recognition. The method was tested by recognizing 10 different activities of assembly line workers in a car production environment.

Selective sampling strategies[5] for activity recognition can adapt dynamically at runtime. Significant savings in energy consumption is achieved via efficient selection of sensor sampling rates for optimal energy-accuracy tradeoff. The optimal selective sampling strategy takes one sample immediately following every transition between activities. This results in an accuracy rate of 100 percent, an average latency of 0 seconds, 0 percent redundancy in the samples taken and 0 percent of missed activities. sampling strategies were developed based on two statistics calculated from the dataset: the distributions of activity durations and the self loop transition probabilities for each activity.

Another sampling stratregy [6] focuses on adaptively varying the sampling rate to achieve the optimal energy consumption above the baseline CPU power consumption. Models were developed to decide whether to perform classification tasks and inference algorithms, locally or remotely. In the first model, such decisions are made in reaction to system changes (e.g., battery levels or foreground processing load). Changes in the user's activities are determining factors in assessing where and to what extent the classification needs to be executed in the second model. Finally, a multi-criteria decision theory is used by the designers of the third model to distribute the computational tasks between a wearable node and the server.Increasing the number of sensor nodes, the sampling frequency of each sensor, and the number of features to extract will all result in improved classification accuracy at the expense of a larger energy overhead. The study makes a connection between two broad research topics, feature selection and power-aware design, that are disjointedly explored in machine learning and embedded system design domains, respectively.

Selecting optimal features according to the maximal statistical dependency criterion based on mutual information[7] obtained satisfying results. Because of the difficulty in directly implementing the maximal dependency condition, an equivalent form was derieved, called minimal-redundancy-maximal-relevance criterion (mRMR), for first-order incremental feature selection. Then, a two-stage feature selection algorithm is presented by combining mRMR and other more sophisticated feature selectors (e.g., wrappers). This allowed selecting a compact set of superior features at very low cost.

A new framework tried to obtain better results using Relief and Correlation Feature selection[8] which are filter methodsfor generation of feature pool for hybrid GA which is a wrapper method using Naïve Bayesian classifier. Thus, combined advantages of both filter and wrapper methods is obtained to get better subset of features and not compromising with the classification accuracy, which is the current need of the hour. Arrays of features is taken chosen by a few feature choice calculations to structure the group. Then the steps of selection, mutation, crossover is continued until the fitness function is satisfied.

III. PROPOSED SYSTEM

The proposed system overcomes the disadvantage of the existing systems by considering the reduction in power consumption without compromising the classification accuracy. It takes into consideration the reduced set of features that are employed to identify a critical event in the specific medical application. The features are selected with the computational cost as weightage. The upcoming section gives a detailed view of deploying the wearable sensors for the application of activity recognition. decision support. This processing chain can be closed by a feedback loop from the back-end server to the user to transmit and receive data.

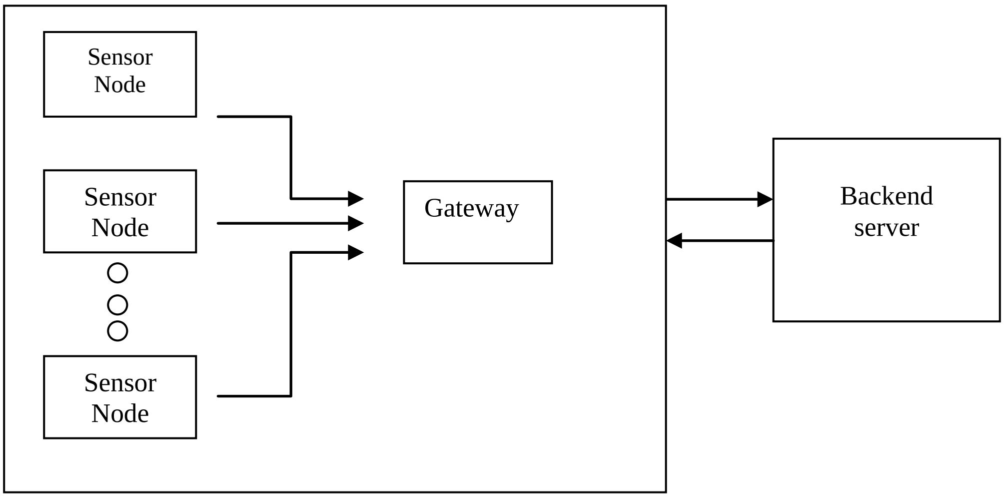


Figure1. Architecture of a wearable sensor network

medical application. The features are selected with the computational cost in concern. The following section gives a detailed view of employing the wearable sensors for the application of activity recognitiondecision support. This processing chain can be closed by a feedback loop from the back-end server to the user the reduced set of features that are employed to identify a critical event in the specific medical application.. A wearable sensor network, also called body sensor network, is composed of several body-worn sensor nodes, a gateway, and a back-end server as shown in Figure.1

Each sensor node is attached to the body to sample and process physiological signals and transmit partial results to the gateway. A sensor node usually has several sensors for capturing different user's states (e.g., body acceleration), an embedded processor to perform limited signal processing and information extraction, and a radio for data transmissions. The gateway is a more powerful unit such as a cell phone or a PDA that performs data fusion and makes conclusions about current state of the user (e.g., 'walking', 'running', and 'sitting'). The results are further transmitted, through the Internet, to a back-end server for storage, further processing, and clinical-

A. Signal Processing

Sensor node in a wearable system processes sensor readings through a chain of embedded signal processing modules, each of which is intended to extract partial information from the signal and reduces the amount of sampled data. Figure 2 illustrates a typical signal processing flow for applications targeting classifying physiological signals into user's states.

In physical movement monitoring applications, readings from motion sensors such as accelerometers, magnetometers, and gyroscopes undergo signal processing to classify human actions such as 'walking', 'sit to stand', and 'jumping'. Signal processing and pattern recognition for wearable sensor networks require a learning phase during which the system is trained based on a training data set. During this phase, parameters of the system used in different signal processing modules are adjusted. Such parameters define the training model.

Feature selection is only part of the learning process. The signals that are sampled by each sensor node are first passed through a filter to reduce high frequency noise. Segmentation is intended to identify 'start' and 'end' points of the actions being classified.

Figure2. Signal Processing

Feature extraction module is responsible for calculating statistical and morphological characteristics of the signal segment. Finally, a classification algorithm is utilized to determine the current state of the user.

B. *Optimal Feature Selection*

The exhaustive set of features involved in activity recognition, may lead to more power consumption which may lead to battery depletion. This is the core part of the proposed system where the actual notion of power-aware comes into play. The flow of optimal feature selection process is depicted in Figure 3.

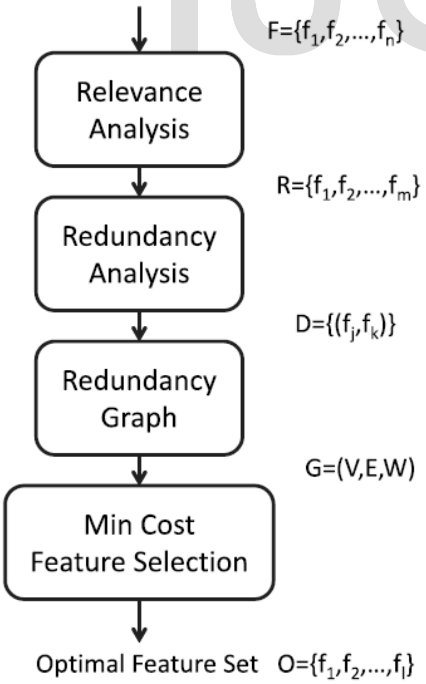


Figure 3 Proposed approach for optimal power-aware feature selection

C. *Relevance Analysis*

The symmetric uncertainty is used to find the relevance between the features and the activities to be recognized. Given an exhaustive set of n features F = { f1, f2, . . fn} and a set of human actions A = {a1, a2, . . . , a_h} to be classified, a feature f_i is irrelevant to the classification task if

$$\min_{j}\left(U\left(f_i, a_j\right)\right) < \lambda_R \qquad \text{--- >(1)}$$

where λ_R (relevance threshold) is a design parameter. U(feature,action) is the symmetric uncertainity. Relevance analysis will eliminate features that are irrelevant to the action recognition. A feature is eliminated only if it is below the relevancy threshold for all the considered actions. Each feature is compared with the entire set of actions and then only the decision is taken. A feature is not eliminated even if it relevant to a single action. The remaining m features (m < n) are subject to redundancy analysis whose main goal is to find strongly correlated features.

D. *Redundancy Analysis*

The symmetric uncertainty is used to find the redundant features. If the redundancy is found, then one feature can be computed using the other, so that one can be eliminated from the optimal feature set. Two features fi and fk are considered to be strongly correlated if

$$U\left(f_i, a_j\right) > \lambda_D \qquad \text{------ >(2)}$$

where λ_D (redundancy threshold) is a design parameter. The output of the redundancy analysis is a set of feature pairs in the form of (fi,fk), which are strongly correlated and either of them can be eliminated according to the correlation analysis. However, these features are further examined for computing complexity using the graph model.

E. Redundancy Graph Formation

The redundancy graph is constructed using the more correlated features. The processing cost attributed to each feature is represented by the weight of each feature. The correlated features forms the connected components in the graph.

Given m relevant features introduced by the relevance analysis and a set of feature pairs {fj,fk} generated according to the redundancy analysis, an undirected graph G=(V, E, W) is called redundancy graph, where V is a set of m vertices, V = {u1, u2, . . . , um} associated with the m relevant features, E = {e1, e2, . . . , er} is the set of r feature pairs that are strongly correlated, and W = {w1, w2, . . . wm} is the set of weights, assigned to the vertices, denoting the computing cost associated with each feature.

F. Greedy Approach

Our greedy algorithm for solving MCFS problem is presented in Algorithm 1. For each vertex ui in the redundancy graph, the algorithm first finds all adjacent vertices (Vi). It then finds the best candidate vertex to include in the final vertex set (V). The best candidate is the one with maximum profit. A maximum profit vertex is the one with maximum value of “cardinality of Vi divided by vertex cost wi”. The intuition behind selecting such a vertex is that it has a large number of adjacent vertices and a small cost. Finally, the algorithm adds the candidate vertex (u_i) to V and eliminates ui and all its neighbours from Vi as well as V. The algorithm iterates until there is no more vertex in V indicating that each vertex is either chosen as a final vertex or is dominated by a final vertex.

ALGORITHM 1:

Require: Redundancy Graph G=(V,E,W)
Ensure: Final vertex set Ω
1. Ω = ∅
2. For all u_i ϵ V do
3. V_i = { all vertices u_j adjacent to u_i }
4. End for
5. While V≠∅ do
6. $V_i \leftarrow argmax_{V_i} \frac{V_i}{w_i}$
7. $\Omega = \Omega \cup \{u_i\}$
8. $V_i \leftarrow V_i \setminus \{u_i, u_j\}$ and $V \leftarrow V \setminus \{u_i, u_j\}$
9. End while

IV. IMPLEMENTATION

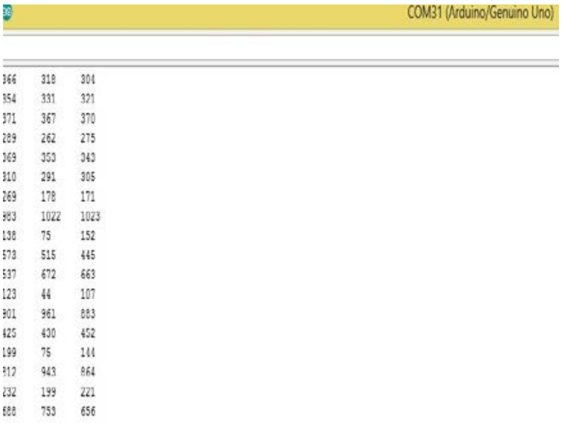
The wearable sensor network for the application of activity recognition includes the sensors, controller and communication components. The sensors are connected to the microcontroller and a zigbee transmitter receiver module is used to communicate the data from the sensors to the PC for further processing of data

The arduino microcontroller provides an IDE package that provides an easy interface to code. The microcontroller is programmed using the IDE and the code is uploaded to the device for execution. When the code gets uploaded, the data can be received from the sensors and are displayed in IDE console.

The accelerometer sensor provides the xyz acceleration values as shown in Figure 4. The subject is asked to perform the specific tasks and the corresponding acceleration data are collected for the specified frequency. The x-y-z pins of the accelerometer is connected with the digital pins of the arduino board to read the acceleration values in the three axes. Since this is the training data used for the further processing, it would be easy if it is stored in a file or a database than acquiring the data everytime. Hence, PLX-DAQ the

ISSN 2229-5518

data acquisition software is used to gather the data from arduino IDE. It is connected with the same port as the arduino controller and the same baud rate is specified.



366	318	304
354	331	321
371	367	370
289	262	275
369	353	343
310	291	305
269	178	171
983	1022	1023
138	75	152
573	515	445
537	672	663
123	44	107
901	961	883
425	430	452
199	75	144
312	943	864
232	199	221
688	753	656

Figure 4. Accelerometer Data

The acquired data is processed using matlab. The preprocessing steps such as noise elimination, normalization and randomization are done for the acquired data as shown in Figure 5. Segmentation is performed by selecting a window size with 50% overlapping. The accelerometer values are influenced by gravity. So the effect of gravitational force is elliminated by seperating the gravitational and body motion components. The gravitational force is assumed to have low frequency. A list of 361 features are calculated for the sampled data where a snapshot is shown in Figure 6.

The feature vector includes both the time domain and frequency domain features. The power consumed for the feature extraction is calculated using the powertop tool available. Then the features are selected based on the optimal algorithm proposed. Inorder to reduce the amount of processing, the window size can be increased because the frequency of change of activity is very less. Decision Tree Classifier is used to classify the physical activity performed with 70:30 ratio for training and test data set.

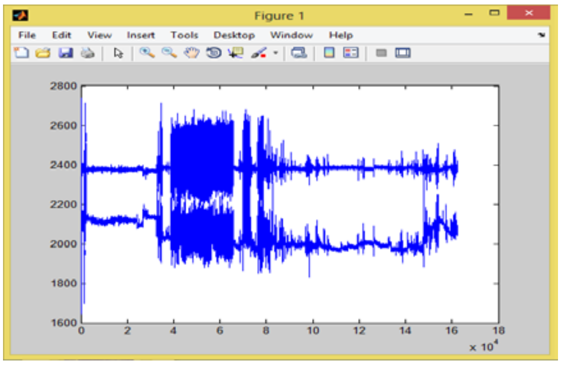
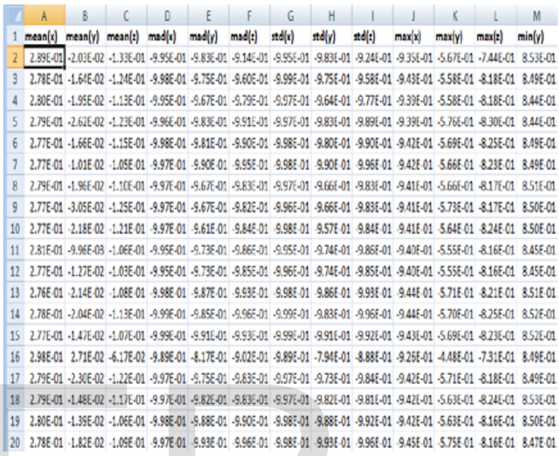


Figure 5. Preprocessed Signal



	A	B	C	D	E	F	G	H	I	J	K	L	M
1	mean(x)	mean(y)	mean(z)	mad(x)	mad(y)	mad(z)	std(x)	std(y)	std(z)	max(x)	max(y)	max(z)	min(y)
2	2.89E-01	-2.03E-02	-1.33E-01	-9.95E-01	-9.83E-01	-9.14E-01	-9.95E-01	-9.83E-01	-9.24E-01	-9.35E-01	-5.67E-01	-7.44E-01	8.53E-01
3	2.78E-01	-1.64E-02	-1.24E-01	-9.98E-01	-9.75E-01	-9.60E-01	-9.99E-01	-9.75E-01	-9.58E-01	-9.43E-01	-5.58E-01	-8.18E-01	8.49E-01
4	2.80E-01	-1.95E-02	-1.13E-01	-9.95E-01	-9.67E-01	-9.79E-01	-9.97E-01	-9.64E-01	-9.77E-01	-9.39E-01	-5.58E-01	-8.18E-01	8.44E-01
5	2.79E-01	-2.62E-02	-1.23E-01	-9.96E-01	-9.83E-01	-9.91E-01	-9.97E-01	-9.83E-01	-9.89E-01	-9.39E-01	-5.76E-01	-8.30E-01	8.44E-01
6	2.77E-01	-1.66E-02	-1.15E-01	-9.98E-01	-9.81E-01	-9.90E-01	-9.98E-01	-9.80E-01	-9.90E-01	-9.42E-01	-5.69E-01	-8.25E-01	8.49E-01
7	2.77E-01	-1.01E-02	-1.05E-01	-9.97E-01	-9.90E-01	-9.95E-01	-9.98E-01	-9.90E-01	-9.96E-01	-9.42E-01	-5.66E-01	-8.23E-01	8.49E-01
8	2.79E-01	-1.96E-02	-1.10E-01	-9.97E-01	-9.67E-01	-9.83E-01	-9.97E-01	-9.66E-01	-9.83E-01	-9.41E-01	-5.66E-01	-8.17E-01	8.51E-01
9	2.77E-01	-3.05E-02	-1.25E-01	-9.97E-01	-9.67E-01	-9.82E-01	-9.96E-01	-9.66E-01	-9.83E-01	-9.41E-01	-5.73E-01	-8.17E-01	8.50E-01
10	2.77E-01	-2.18E-02	-1.21E-01	-9.97E-01	-9.61E-01	-9.84E-01	-9.98E-01	-9.57E-01	-9.84E-01	-9.41E-01	-5.64E-01	-8.24E-01	8.50E-01
11	2.81E-01	-9.96E-03	-1.06E-01	-9.95E-01	-9.73E-01	-9.86E-01	-9.95E-01	-9.74E-01	-9.86E-01	-9.40E-01	-5.55E-01	-8.16E-01	8.45E-01
12	2.77E-01	-1.27E-02	-1.03E-01	-9.95E-01	-9.73E-01	-9.85E-01	-9.96E-01	-9.74E-01	-9.85E-01	-9.40E-01	-5.55E-01	-8.16E-01	8.45E-01
13	2.76E-01	-2.14E-02	-1.08E-01	-9.98E-01	-9.87E-01	-9.93E-01	-9.98E-01	-9.86E-01	-9.93E-01	-9.44E-01	-5.71E-01	-8.21E-01	8.51E-01
14	2.78E-01	-2.04E-02	-1.13E-01	-9.99E-01	-9.85E-01	-9.96E-01	-9.99E-01	-9.83E-01	-9.96E-01	-9.44E-01	-5.70E-01	-8.25E-01	8.52E-01
15	2.77E-01	-1.47E-02	-1.07E-01	-9.99E-01	-9.91E-01	-9.93E-01	-9.99E-01	-9.91E-01	-9.92E-01	-9.43E-01	-5.69E-01	-8.23E-01	8.52E-01
16	2.98E-01	2.71E-02	-6.17E-02	-9.89E-01	-8.17E-01	-9.02E-01	-9.89E-01	-7.94E-01	-8.88E-01	-9.26E-01	-4.48E-01	-7.31E-01	8.49E-01
17	2.79E-01	-2.30E-02	-1.22E-01	-9.97E-01	-9.75E-01	-9.83E-01	-9.97E-01	-9.73E-01	-9.84E-01	-9.42E-01	-5.71E-01	-8.18E-01	8.49E-01
18	2.79E-01	-1.48E-02	-1.17E-01	-9.97E-01	-9.82E-01	-9.83E-01	-9.97E-01	-9.82E-01	-9.81E-01	-9.42E-01	-5.63E-01	-8.24E-01	8.53E-01
19	2.80E-01	-1.39E-02	-1.06E-01	-9.98E-01	-9.88E-01	-9.90E-01	-9.98E-01	-9.88E-01	-9.92E-01	-9.42E-01	-5.63E-01	-8.16E-01	8.50E-01
20	2.78E-01	-1.82E-02	-1.09E-01	-9.97E-01	-9.93E-01	-9.96E-01	-9.98E-01	-9.93E-01	-9.96E-01	-9.45E-01	-5.75E-01	-8.16E-01	8.47E-01

Figure 6 Feature Vector

V CONCLUSION

Whereas the first decade of research in the field of wearable technology was marked by an emphasis on the engineering work needed to develop wearable sensors and systems, recent studies have been focused on the application of such technology toward monitoring health and wellness. The accuracy and power tradeoffs in wearable sensor networks have been investigated in order to guarantee classification accuracy, while minimizing the system's power consumption. The primary focus was on the feature selection in the wearable sensor networks. Future enhancement can be done by dynamically selecting the nodes for processing based on the activity performed.

REFERENCES

[1] A. Pantelopoulos and N. Bourbakis, “A survey on wearable sensor-based systems for health monitoring and prognosis,” IEEE Trans. Syst., Man, Cybern., Part C: Appl. Rev., Jan. 2010.

[2] H. Ghasemzadeh and R. Jafari, “Coordination analysis of human movements with body sensor networks: A signal processing model to evaluate baseball swings,” IEEE Sens. J., vol. 11, Mar. 2011.

[3] B. Zhao, H. Aydin, and D. Zhu, “Generalized reliability oriented energy management for real-time embedded applications,” in Proc. 48th ACM/EDAC/IEEE Design Autom. Conf., 2011.

[4] P. Zappi, C. Lombriser, T. Stiefmeier, E. Farella, D. Roggen, L. Benini, and G. Troster, “Activity recognition from on-body sensors: Accuracy-power trade-off by dynamic sensor selection,” in Proc. 5th Eur. Conf. Wireless Sensor Netw., vol. 4913,2008.

[5] B. French, D. Siewiorek, A. Smailagic, and M. Deisher, “Selective sampling strategies to conserve power in context aware devices,” in Proc. 11th IEEE Int. Symp. Wearable Comput., pp. 77–80,2007.

[6] Z. Yan, V. Subbaraju, D. Chakraborty, A. Misra, and K. Aberer, “Energy efficient continuous activity recognition on mobile phones: An activity-adaptive approach,” in Proc. 16th Int. Symp. Wearable Comput., pp. 17–24,2012.

[7] Hanchuan Peng, Fuhui Long, and Chris Ding, “Feature Selection Based on Mutual Information: Criteria of Max-Dependency, Max Relevance and Min-Redundancy”, IEEE transactions on pattern analysis and machine intelligence, vol. 27, no. 8, page(s) 154-158, 2005.

[8] Suruchi Koul, Rita Chhikara,” A Hybrid Genetic Algorithm to Improve Feature Selection” on International Journal of Engineering Research & Technology (IJERT) ISSN: 2278-0181 IJERTV4IS050496 vol. 4 Issue 05,2015.