An Energy-Efficient Computational Model for Uncertainty Management in Dynamically Changing Networked Wearables

Ramyar Saeedi, Ramin Fallahzadeh, Parastoo Alinia, Hassan Ghasemzadeh
School of Electrical Engineering and Computer Science
Washington State University, Pullman, WA 99164–2752
{rsaeedi, rfallahz, palinia, hassan}@eecs.wsu.edu

ABSTRACT

The utility of wearables is currently limited to lab experiments and controlled environments mainly because computational algorithms embedded in wearables fail to produce accurate measurements in uncontrolled, dynamically changing, and potentially harsh environments. With the exponentially growing adoption of these systems in human-centered Internet-of-Things (IoT) applications, development of resource-efficient solutions to enhance the accuracy of this systems remains a considerable research challenge. In this paper, we introduce an energy-efficient framework for uncertainty management of networked wearables. The core components of our framework are anomaly screening units for detecting anomalies that require handling, thus resulting in one order of magnitude less energy consumption compared to the conventional frameworks. Furthermore, our screening approach achieves 98.3% accuracy in detecting anomalies based on real data collected with wearable motion sensors.

CCS Concepts

• General and reference → Reliability; Performance; • Human-centered computing → Ubiquitous and mobile computing systems and tools; • Computing methodologies → Feature selection; • Computer systems organization → Embedded software; • Hardware → Emerging architectures;

Keywords

Networked wearables, accelerometer, activity monitoring, power optimization, reliability, uncertainty management, feature selection

1. INTRODUCTION

Networked wearables have gained tremendous attention recently due to their potential to enable emerging applications in wellness, health-care, fitness as one of the main focus in Internet-of-Things (IoT) domains [1][3]. The utilization of these systems, however, is currently limited to controlled environments, lab settings, and small field trials. This limitation has created major obstacles in scaling these systems up and advancing their utility in real-world settings.

Wearables collect sensors data and monitor/predict events of interest (e.g., daily activities), which often requires development of machine learning algorithms for real-time execution. Examples of such algorithms are classification, template matching, and clustering. When used in real world scenarios, wearables face different types of anomaly, such as mis-orientation, displacement, and misplacement of sensors, which dramatically impact performance. For example, one can train a classifier, based on a training dataset, to perform activity recognition. The classification algorithm, however, fails to detect physical activities, if the location of the on-body sensors changes [7]. Therefore, it is crucial to develop uncertainty management frameworks that detect and mitigate the effect of such anomalies on the performance of the machine learning algorithms.

Current approaches for uncertainty management events are significantly energy-inefficient: (1) these techniques aim to mitigate the effects of a particular anomaly (e.g., mis-orientation) on a specific application of wearables (e.g., activity recognition) [1][3][5]. As a result, as shown in Figure 1, these techniques require separate unit for each anomaly; (2) although anomalies are sparse, occurring at most 10% of the times [7], current algorithms require constant execution of several anomaly management units (AMUs), an approach that demands extremely high computational power, given the stringent constraint resources of wearables.

The aforementioned drawbacks with the current uncertainty management approaches warrant development of a new class of models that not only are accurate in detecting anomalies and mitigating the effect of such events on the performance but also are energy-efficient. To the best of our knowledge, development of energy-
In this paper, we present an approach to construct a cost-efficient feature selection model for anomaly detection. This approach identifies a set of redundant features with minimum computation and communication cost. We first introduce a new metric, called prediction index, which uses the inter-feature correlation and estimates the potential power of two features for predicting an anomaly. For this purpose, it is needed to have training controlled data and data in presence of anomaly as shown in Figure 3. The prediction index is then used to construct a graph model, called Prediction Graph, which is further employed to find a set of feature pairs in constructing SUs. For each anomaly type, we construct a specific prediction graph. Finally, a set of energy efficient features are selected from each prediction graph to build individual screening units which together construct the anomaly screening unit (ASU).

3. ANOMALY SCREENING UNIT

In this section, we present our cost-efficient feature selection approach to construct the computational model of SUs to detect anomaly. This approach identifies a set of redundant features with minimum computation and communication cost. We first introduce a new metric, called prediction index, which uses the inter-feature correlation and estimates the potential power of two features for predicting an anomaly. For this purpose, it is needed to have training controlled data and data in presence of anomaly as shown in Figure 3. The prediction index is then used to construct a graph model, called Prediction Graph, which is further employed to find a set of feature pairs in constructing SUs. For each anomaly type, we construct a specific prediction graph. Finally, a set of energy efficient features are selected from each prediction graph to build individual screening units which together construct the anomaly screening unit (ASU).

3.1 Problem Statement

Let $S = \{s_1, s_2, \ldots, s_n\}$ be a set of ‘n’ sensor nodes forming a networked wearable system. Furthermore, let $F_1 = \{f_{i1}, f_{i2}, \ldots, f_{i\text{m}}\}$ be a set of ‘m’ features extracted from sensor node $s_i$.

**DEFINITION 1 (PREDICTION INDEX).** Given two features $f_{ij}$ and $f_{ik}$, prediction index is defined as

$$
\chi_{ij}^{kl} = \frac{U_R(f_{ij}, f_{ik})}{U_A(f_{ij}, f_{ik})}
$$

where $U_R$ denotes correlation between the two features during normal operation of the network while $U_A$ represents inter-feature correlation associated with presence of an anomaly. In this paper, we use symmetric uncertainty for correlation measurement. Symmetric uncertainty captures non-linear correlation between variables and therefore, is a safe measure for feature analysis studies.

**DEFINITION 2 (SYMMETRIC UNCERTAINTY).** The symmetric uncertainty between two discrete random variables $X$ and $Y$ is given by:

$$
U(X,Y) = \frac{2 \times I(X,Y)}{H(X) + H(Y)}
$$
where \( H(X) \) and \( H(Y) \) represent the entropy of random variables \( X \) and \( Y \), respectively, and \( I(X, Y) \) denotes the information gain between the two variables. \( I(X, Y) \) is defined by:
\[
I(X, Y) = H(X) - H(X|Y)
\]

(3)

Symmetric uncertainty is the normalized information gain and is always between 0 and 1, where \( U=1 \) means that knowing the value of either variable can completely predict the other variable, and \( U=0 \) indicates that the two variables are completely independent.

DEFINITION 3 (STRONG PAIR FEATURES). Two features \( f_{ij} \) and \( f_{kl} \) are considered as a strong pair, if the prediction index associated with them exceeds a given threshold, \( \Delta \):
\[
\lambda_{ij}^{kl} \geq \Delta
\]

(4)

DEFINITION 4 (PREDICTION GRAPH). The prediction graph \( G = (V, E, C_V, W_{E}) \) is an undirected graph with the following specifications. The variable \( V \) denotes the set of \( m \times n \) vertices associated with the features extracted from the sensor nodes in \( S \): \( E \) represents the set of edges such that an edge \( e_{ij} \) exists if features \( f_{ij} \) and \( f_{kl} \) form a strong pair as defined previously. The set \( C_V \) represents vertex weights. A weight \( c_{ij} \) is assigned to each vertex \( v_{ij} \) to represent the cost of feature \( f_{ij} \) computation. Furthermore, \( W_{E} \) represents edge weights. Each edge \( e_{ij} \) has a cost \( w_{ij} \) given by
\[
w_{ij} = \begin{cases} 
\beta, & \text{if } i \neq k \\
0, & \text{otherwise}
\end{cases}
\]

(5)

where \( \beta \) is the cost of data transmission between two different sensor nodes \( s_i \) and \( s_k \).

PROBLEM 1 (MIN-COST REDUNDANT FEATURE SELECTION). Min-Cost Redundant Feature Selection (MCRFS) is to select a subset of features such that the total cost in terms of computation and communication is minimized, while we have at least one feature from each sensor node.

Given that wearables are used in a highly dynamic environments, the measurements are likely to be noisy, which may reduce the effectiveness of feature pairs in predicting anomalies. In order to further enhance robustness of the system, our proposed framework provides flexibility in reliability level of feature selection approach, which select different set of features. Motivated by K-connectivity in wireless sensor networks, we define K-reliable feature selection as follows [10].

PROBLEM 2 (K-RELIABLE FEATURE SELECTION). Given an exhaustive set of features, K-reliable feature selection is defined as finding \( K \) set of features which meet the conditions of our framework. In each iteration, the selected edges are removed from prediction graph for the next round.

3.2 Problem Formulation

The MCRFS problem can be formulated as follows. Assume \( a_{ij}^{kl} \) is a binary variable indicating whether or not the edge \( e_{ij} \) exists in the prediction graph \( G \). That is:
\[
a_{ij}^{kl} = \begin{cases} 
1, & \text{if } e_{ij} \in E \\
0, & \text{otherwise}
\end{cases}
\]

(6)

The goal is to choose a subset of the edges such that all sensor nodes have at least one feature candidate in the final feature set. By choosing an edge from the prediction graph, we can perform a cross-feature analysis to determine whether or not the system has encountered an uncertainty event. Let \( x_{ij}^{kl} \) be a binary variable indicating whether or not the edge \( e_{ij} \) is selected by our feature selection algorithm. Thus,
\[
x_{ij}^{kl} = \begin{cases} 
1, & \text{if } e_{ij} \text{ is selected} \\
0, & \text{otherwise}
\end{cases}
\]

(7)

Thus, the optimization problem can be written as follows.
\[
Z = \sum_{i,k=1}^{n} \sum_{j,l=1}^{n} a_{ij}^{kl}(c_{ij} + w_{ij}^{kl} + c_{kl})x_{ij}^{kl}
\]

(8)

Minimize \( Z \)

Subject to:
\[
\sum_{i,k=1}^{n} \sum_{j,l=1}^{n} x_{ij}^{kl} \geq 1, \quad \forall i \in \{1, \ldots, n\}
\]

(9)

\[
x_{ij}^{kl} \in \{0, 1\}, \quad \forall i, k, j, l
\]

(10)

THEOREM 1. cc MCRFS problem is NP-hard and can not be approximated for any time computable function \( \alpha(t) \).

Proof A reduction from the Weighted Set Cover (WSC) problem to MCRFS is used to prove NP-hardness. Consider the universe \( U = \{s_1, s_2, \ldots, s_n\} \) and a collection of sets \( S = \{(e_1, w_{e_1}), (e_2, w_{e_2}), \ldots, (e_k, w_{e_k})\} \) along with their weights. If we repeatedly add several copies of each set to the collection of sets, the WSC problem will not change in terms of complexity. Also, this new problem is equivalent to the MCRFS, if we consider each edge as a set. Each set contains those nodes in the network which are connected by the corresponding edge. Several replica for each set is because of having super-nodes(network nodes), which have multiple connections between them. As this problem is a reduction of WSC problem, the MCRFS problem is an NP-Hard problem.

In the next section, we present a greedy approximation algorithm for solving the problem in polynomial time.

3.3 Greedy Algorithm

Algorithm 1 shows the proposed greedy algorithm. The algorithm keeps track of selected features in \( F \). The function \( h(F) \) outputs the sensor nodes that are associated with a given feature set \( F \). During each iteration, the algorithm chooses an edge with minimum cost per each new covered sensor node by that edge. The set of newly covered sensor nodes by edge \( e_{ij} \) is computed by \( h(F \cup \{s_i, s_k\}) - h(F) \). The algorithm iterates until all sensor nodes are covered (i.e., \( h(F) = h(V) \)). For the case of K-Reliability, the algorithm will repeat the procedure on the graph without the edges that has been selected in previous rounds.

THEOREM 2. The greedy algorithm returns a feature set of at most \( 1.5 \) times the minimum cost of any feature set covering all the sensor nodes.

Proof When the greedy algorithm chooses an edge \( e_{ij} \), imagine that it charges the price per sensor node for that iteration to each sensor node newly covered by \( e_{ij} \). Then the total cost of the edges chosen by the algorithm equals the total amount charged, and each element is charged only once. Since features are chosen in pairs, consider any pair of features \( f_{ij} \) and \( f_{kl} \) in the optimal feature set, \( F^* \) associated with edge \( e_{ij} \).

During the iteration that \( e_{ij} \) is chosen, one (i.e., either \( s_i \) or \( s_j \)) or two (both \( s_i \) and \( s_j \)) are uncovered yet. If only one sensor node is
uncovered, then the greedy algorithm would pay a cost per sensor node of at most $W_{ij}^{kl}$ charged to the sensor node being covered during this iteration ($W_{ij}^{kl} = c_{ij} + w_{ij}^{cl} + c_{kl}$). If, however, two sensor nodes are uncovered at the beginning of the iteration, then the cost per sensor node would be the sum of cost per sensor node of at most $W_{ij}^{kl}/2$. Summing over all sensor nodes associated with $c_{ij}^{kl}$, the total amount charged is at most $W_{ij}^{kl} + W_{ij}^{kl}/2 = 1.5 \times W_{ij}^{kl}$. Summing over all edges in $F^*$ and noting that every sensor node is covered by some edge in $F^*$, the total amount charged to sensor nodes overall is given by

$$\sum_{W_{ij}^{kl} \in F^*} 1.5 \times W_{ij}^{kl} = 1.5 \times OPT$$

(12)

4. VALIDATION

4.1 Energy Model

We define energy models for both conventional and our framework as shown in Figure 1 and Figure 2, respectively. The energy consumption of a wearable sensor node consists of three components: (1) data sampling; (2) feature extraction; and (3) machine learning algorithms. According to Figure 1, the total energy consumption can be calculated by

$$E(\text{conv}) = E(\text{ml}) + E(s) + \sum_{k=1}^{N} E(AMU_k)$$

(13)

where $E(\text{conv})$ represents energy consumption of the conventional models, $E(\text{ml})$ represents energy consumption of the application algorithm (e.g., activity recognition), $E(s)$ denotes energy consumption due to data sampling, and $E(AMU_k)$ is energy consumption of the $k$-th anomaly management unit.

The total energy consumption of our uncertainty management framework, shown in Figure 2, can be written as:

$$E(\text{scr}) = E(\text{ml}) + E(s) + \sum_{k=1}^{N} \alpha_k \times E(AMU_k) + \sum_{i=1}^{M} E(SU_i)$$

(14)

where $E(\text{scr})$ denotes energy consumption of the proposed model based on anomaly screening, $E(SU_i)$ represents energy consumption of the $i$-th screening unit, and $\alpha_k$ indicates the activation frequency for $SU_k$ and is given by

$$\alpha_k = (F_k \times tp_k + (1 - F_k) \times fp_k)$$

(15)

where $tp_k$ and $fp_k$ denote true positive rates and false positive rates of the $SU_k$ classifier respectively, and $F_k$ represents the frequency of occurrence of the $k$-th anomaly type. These rates indicates how often a mitigation unit ($MU_k$) need to be activated by corresponding $SU_k$. We note that, in this formulation, the energy consumption due to feature extraction is considered as a part of the processing energy consumption for AMU, SU and MU units.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
<th>Energy (nJ)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amp</td>
<td>Amplitude of signal segment</td>
<td>16386</td>
</tr>
<tr>
<td>Med</td>
<td>Median of signal segment</td>
<td>405159</td>
</tr>
<tr>
<td>Mean</td>
<td>Mean value of signal segment</td>
<td>8126</td>
</tr>
<tr>
<td>Max</td>
<td>Maximum amp of signal segment</td>
<td>8103</td>
</tr>
<tr>
<td>Min</td>
<td>Minimum amp of signal segment</td>
<td>8129</td>
</tr>
<tr>
<td>P2P</td>
<td>Peak to peak amplitude</td>
<td>16291</td>
</tr>
<tr>
<td>Var</td>
<td>Variance of signal segment</td>
<td>38846</td>
</tr>
<tr>
<td>Std</td>
<td>Standard deviation</td>
<td>40431</td>
</tr>
<tr>
<td>RMS</td>
<td>Root mean square power</td>
<td>29705</td>
</tr>
<tr>
<td>S2E</td>
<td>Start to end value</td>
<td>83</td>
</tr>
<tr>
<td>Morph</td>
<td>Morphological Samples</td>
<td>41</td>
</tr>
</tbody>
</table>

4.2 Experimental Setup

To demonstrate the effectiveness of the proposed framework, we constructed a network of three wearables placed on 'left wrist', 'right ankle', and 'waist'. We used TelosB motes to measure acceleration and angular velocity of different body segments using a triaxial accelerometer and a biaxial gyroscope. Three subjects were asked to perform 30 different daily activities such as 'sit to stand', 'sit to lie', 'bend and grasp', 'kneeling', 'step forward', 'step backward', etc. Each transitional movement was repeated 10 times by each subject. The sensor nodes were programmed to sample data at 50Hz and transmit the data wirelessly to a laptop computer. In addition to these locations, movement data were collected for several other network configurations. These configurations resemble anomaly (i.e. misplacement) that can happen in real scenarios. We here includes sensor misplacement and displacement for our case study.

We extracted 10 statistical features as well as 10 morphological features from each activity trial. The features are shown in Table 1, where the energy consumption of computing each feature is also reported. We calculated instruction level energy consumption of MSP430 micro-controller, which is available on the TelosB motes used in our experiments. The details of energy calculations is based on an instruction level energy model provided in [9, 11]. Also, information on power consumption and specifications of the sensor nodes can be found in [12]. Morphological features were equally-spaced samples of the activity signal. Overall, we extracted 300 features from three sensors.

For allocating the communication costs in the prediction graph, we assume that the communication cost is zero for edges that connect feature of the same sensor node which runs the uncertainty management. The communication costs of inter-node data transmission was set to be a fixed number computed based on the energy cost of transmitting on data packet using ZigBee protocol. That is, the communication power is based on ZigBee protocol specifications [13].
4.3 Results

In this section, we present our results on (1) the impact of uncertainty events on accuracy performance of activity recognition; (2) performance of individual SUs; and (3) energy efficiency of the proposed uncertainty management framework.

4.3.1 Impact of Anomaly on Accuracy

To show the impact of anomaly on the performance of networked wearables, we used the collected dataset for activity recognition where we develop a machine learning classifier to detect physical activities of daily life. Given that $k$-Nearest Neighbor ($k$NN) classification has shown promising results in activity recognition [4], we chose $k$NN for activity classification. First, we developed a machine learning classifier using the perfect data, and then tested this model using ten-fold cross validation, which provided us with an average accuracy of 97.8% for detecting the 30 activities. Later, we test the activity recognition model on data with different types of anomalies as listed in Table 2. In each case, the assumption is that there is just one anomaly in the network. Table 2 shows the amount of accuracy drop due to these anomaly occurrences, namely sensor misplacement and sensor displacement. As shown in Table 2, the accuracy of activity recognition drops by 22.6% on average.

Table 2: On-body locations used for data collection in normal, and in presence of anomaly, and corresponding accuracy drops.

<table>
<thead>
<tr>
<th>ID</th>
<th>Uncertainty</th>
<th>Type</th>
<th>Drop in Recall</th>
<th>Drop in Precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Right Ankle $\rightarrow$ Left Forearm</td>
<td>Misplacement</td>
<td>36.1%</td>
<td>33.2%</td>
</tr>
<tr>
<td>2</td>
<td>Left Wrist $\rightarrow$ Right Knee</td>
<td>Misplacement</td>
<td>33.4%</td>
<td>28.2%</td>
</tr>
<tr>
<td>3</td>
<td>Waist $\rightarrow$ Neck</td>
<td>Misplacement</td>
<td>25.1%</td>
<td>28.1%</td>
</tr>
<tr>
<td>4</td>
<td>Right Ankle $\rightarrow$ Right Shin</td>
<td>Displacement</td>
<td>32.1%</td>
<td>34.5%</td>
</tr>
<tr>
<td>5</td>
<td>Left Wrist $\rightarrow$ Right Forearm</td>
<td>Displacement</td>
<td>14.2%</td>
<td>16.4%</td>
</tr>
<tr>
<td>6</td>
<td>Waist $\rightarrow$ Left Hip</td>
<td>Displacement</td>
<td>15.8%</td>
<td>16.3%</td>
</tr>
</tbody>
</table>

4.3.2 Performance of Anomaly Screening Unit

To assess the performance of the anomaly screening module, we constructed two prediction graph with 300 vertices associated with the extracted features for two anomaly type. The inter-feature correlations is needed for both perfect data and data in presence of anomaly (i.e. one sensor is misplaced) to build each prediction graph.

The number of edges in the prediction graphs is directly dependent on the value of $\Delta$. Clearly, the number of edges decreases as the threshold increases. We experimentally determined a range of $3.5 \leq \Delta \leq 3.9$ to be an acceptable range. If we choose $\Delta$ outside of these range the graph would be either sparse, or contain many irrelevant features. As the level of reliability increased from $K=1$ to $K=4$, the number of selected features increased from 3 to 9.

To compute the accuracy performance of the screening units that reside in our anomaly screening module, we considered single-anomaly that are listed in Table 2. We used three classification algorithms including Decision Tree (DT), ($k$NN), and support vector machine (SVM) for this analysis. The accuracy of screening units ranged from 73.7% to 100% using different classifiers and different levels of reliability as shown in Figure 4. The average accuracy for misplacement, and displacement cases are 91.1%, and 98.5% using a decision tree classifier.

Table 3 shows the amount of energy savings obtained using our power-aware feature selection technique compare to the baseline scenario. The energy savings ranged from 55.1% for the highest reliability level ($K=4$) to 97.1% for the lowest reliability level ($K=1$) in terms of feature computation power. The communication power saving is also based on the percentage of features that are needed to transmit using the proposed framework compared to the baseline scenario.

4.3.3 Energy Analysis of the Framework

In this section, energy analysis of the propose framework compared to the conventional approach is presented. Given that we focused on two anomaly types, in this paper, our framework consists of two SUs and two MUs one for each anomaly type. Table 3 shows the energy consumption, and accuracy of anomaly detection and mitigation units that are used in our system. The MU units in Table 3 are node localization models which are used to mitigate the effects of misplacement, and displacement.

We measured the total computation energy of both approaches as a function of $\alpha$, the activation rate of the MUs. In this comparison, we only considered energy consumption of signal processing and feature extraction as other energy components are used in both conventional model and proposed framework.

The energy consumption of the two models (i.e., conventional and proposed) is shown in Figure 4a and Figure 4b. With $\alpha = 0.1$, the amount of power savings of our framework is 88% and 69% for $k$NN and DT classifiers respectively. It is reasonable to assume $\alpha = 0.1$ as an uncertainty rate [7]. In short, our system outperforms the conventional model for $\alpha \leq 0.93$ using kNN, and $\alpha \leq 0.81$.
5. CONCLUSION AND FUTURE WORK

We proposed an energy-efficient framework which is used to increase the robustness of wearable sensors. A feature selection algorithm was developed to exploit correlation variation of pair of features. The energy saving of framework is 88.2% and 69.7% compared to conventional methods considering anomaly occurrence of 10% for kNN-based, and DT-based MUs respectively. There is a trade off between power consumption and accuracy of MUs, the more accurate classifier, the lower amount of power saving, simple SU units detect anomaly types with 98.3% accuracy on average.

We focused on activity recognition as our pilot application and performed our analysis on motion sensors such as accelerometers and gyroscope sensors. Our ongoing research involves several expansions with regards to the framework validation. First, we are working on integrating additional sensor modalities such as audio signals and Electromyography (EMG) sensors into our wearable network. Second, we plan on creating a larger sensor network with an exhaustive set of sensors covering many body joints involved in building human skeleton model.

In this paper, we only considered single-fault model where we assumed that each sensor will experience at most one anomaly at a time. Given that anomalies are sparse, this assumption is intuitively reasonable. Yet, the simultaneous occurrence of multiple anomalies is theoretically possible (i.e. multiple sensors, multiple anomaly). Thus, we plan to enhance the system which can deal with multiple-anomaly scenarios as part of our future work.

6. REFERENCES


