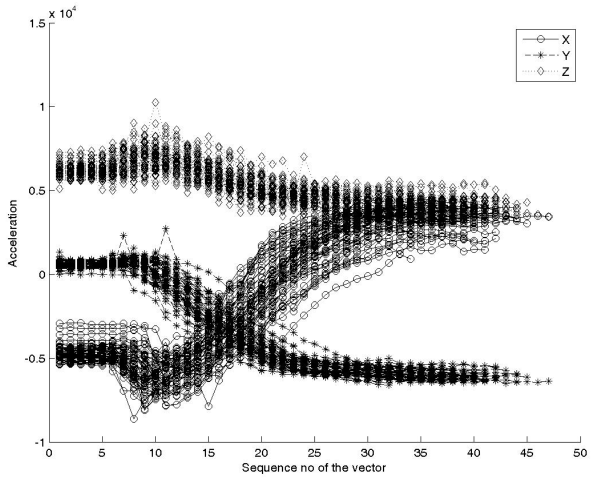
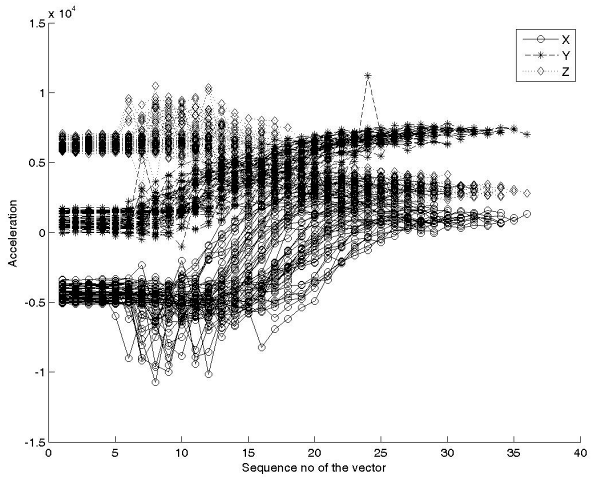
Appendix

A. Supplementary Figures and Equations



(a) Subject 1



(b) Subject 2

Fig. 11: Sensor data for training instances of target activity

$$eval(PV^{(d+1)}, seq, d) = \sum_{i=1}^{d+1} PV_{i-1}^{(d+1)} seq^{i-1} \tag{6}$$

$$\begin{aligned} \forall \alpha, \beta, (r(\alpha) = r(\beta) \implies \alpha = \beta) \\ r(a) = Q_i^{|A|}, \forall a \in A \end{aligned} \tag{7}$$

$$h = \sqrt{n_a} \tag{8}$$

B. Supplementary Algorithms

Algorithm 5 $dist(V^{fd}, seq, DS^{fd})$

```
for $\forall 1 \le i \le fd$ do
    $B_i^{fd} \leftarrow eval(DS_i^{fd}.coef, seq, DS_i^{fd}.deg)$
    if $V_i^{fd} \neq 0$ then
        $I_i^{fd} \leftarrow |trunc(B_i^{fd})|\%|trunc(V_i^{fd})|$
    else
        $I_i^{fd} \leftarrow 0$
    end if
end for
$|\overrightarrow{I}^{fd}| \leftarrow \sqrt{\sum_{i=1}^{fd} (I_i^{fd})^2}$
return $|\overrightarrow{I}^{fd}|$
```

Algorithm 6 majorityVoting-$\Lambda(\Lambda^{n_a})$

```
$H_i^{|A|} \leftarrow 0, \forall 1 \le i \le |A|$
for $\forall 1 \le i \le n_a$ do
    //loop_j:
    for $\forall 1 \le j \le |A|$ do
        if $\Lambda_i^{n_a} = Q_j^{|A|}$ then
            $H_j^{|A|} \leftarrow H_j^{|A|} + 1$
            Quit loop-j
        end if
    end for
end for
$i\_max \leftarrow argmax_{i \in X} f(i)$
$\omega \leftarrow \sum_{i=1}^{|A|} [f(i) = f(i\_max)]$
if $\omega > 1$ then
    return -1
else if $\omega == 1 \wedge i\_max \neq r(a)$ then
    return -1
else
    return $i\_max$
end if
```

Algorithm 6 explains $majorityVoting - \Lambda$ used in Algorithm 3. $majorityVoting - \Lambda$ processes Λ^{n_a}, which is the vector of candidate decisions to find out the simple action a obtaining most of the votes. In Algorithm 6, the elements of the vector $H^{|A|}$ storing the number of votes for each action are set to zero initially. Each element of Λ^{n_a} is compared against the unique code values stored in $Q^{|A|}$ to determine the activity type located in $\Lambda_i^{n_a}$. In the case of a match, the element of $H^{|A|}$ corresponding to the action whose unique code is $Q_j^{|A|}$ is incremented by one, indicating that the action represented by $Q_j^{|A|}$ gains one more vote. Then, $loop\_j$, which iterates as many times as $|A|$, is quit. Let $X = \{1, 2, ..., |A|\}$, $Y = \left\{H_1^{|A|}, H_2^{|A|}, ..., H_{|A|}^{|A|}\right\}$ and $f : X \Rightarrow Y$ such that $f(i) = H_i^{|A|}$. After all elements in $H^{|A|}$ are computed, the element $i \in X$ where $f(i)$ takes its maximum is computed and stored in i_max. The number of occurrences of the simple action whose unique code corresponds to i_max is assigned to ω. Depending on ω, the ultimate decision generated by $majorityVoting - \Lambda$ is found. If ω is greater than 1, then there is more than one candidate simple action which obtains maximum number of votes. In this case, $majorityVoting - \Lambda$ returns -1, indicating a decision cannot be made. If ω is 1 but i_max is not equal to $r(a)$, $majorityVoting - \Lambda$ similarly returns -1. i_max differing from $r(a)$ demonstrates that the simple action obtaining maximum number of votes is different from the simple action whose training samples yield the polynomials used to generate the decisions stored in Λ^{n_a}. If neither of the conditions leading to the *cannot decide* cases mentioned above, i_max is returned signalling the simple action whose unique code corresponds to i_max obtains most of the votes.

Algorithm 7 majorityVoting-D($D^{|A|}$)

```
$H_i^{|A|} \leftarrow 0, \forall 1 \leq i \leq |A|$
**for** $\forall 1 \leq i \leq |A|$ **do**
    //loop_j:
    **for** $\forall a \in A$ **do**
        **if** $D_i^{|A|} = Q_{r(a)}^{|A|}$ **then**
            $H_{r(a)}^{|A|} \leftarrow H_{r(a)}^{|A|} + 1$
            Quit loop-j
        **end if**
    **end for**
**end for**
$i\_max \leftarrow argmax_{i \in X} f(i)$
$\omega \leftarrow \sum_{i=1}^{|A|} [f(i) = f(i\_max)]$
$\zeta \leftarrow \sum_{i=1}^{|A|} [f(i) = -1]$
**if** $\omega > 1$ **then**
    return -1
**else**
    **if** $H_{i\_max}^{1x|A|} < \zeta$ **then**
        return -1
    **else**
        return $i\_max$
    **end if**
**end if**
```

$majorityVoting - D$ shown in Algorithm 7 is similar to Algorithm 6, however, the structure of the vectors on which they operate are different from each other, consequently, the majority voting scheme presented in $majorityVoting - D$ exhibits some differences from that in $majorityVoting - \Lambda$. While $majorityVoting - \Lambda$ targets finding the most frequently occurring label in Λ^{n_a}, $majorityVoting - D$ performs the same task on $D^{|A|}$. Each element of Λ^{n_a} corresponds to a decision output by $cls\_prediction$ when applied on Ω_R and η, hence $\Lambda_i^{n_a} \in \{Q_1^{|A|}, Q_2^{|A|}, ..., Q_{|A|}^{|A|}\}, \forall 1 \leq i \leq n_a$. On the other hand, each element of $D^{|A|}$ corresponds to the decision returned by $majorityVoting - \Lambda$ applied on Λ^{n_a}, therefore, $D_i^{|A|} \in \{Q_1^{|A|}, Q_2^{|A|}, ..., Q_{|A|}^{|A|}\} \vee D_i^{|A|} \in \{-1\}$, where -1 signals that Λ^{n_a} does not yield a decision while being processed by $majorityVoting - \Lambda$ as mentioned in Algorithm 6. In Algorithm 7, ω is computed as in Algorithm 6, additionally, ζ, which is the number of occurrences of -1 in $H^{|A|}$. For the unique code i_max of an action $a \in A$ to be returned as the decision, the maximum number of votes obtained by a, which is $H_{i\_max}^{|A|}$, should surpass ζ as well as satisfying the criterion of ω being greater than 1.

Algorithm 8 training_compared()

```
$\Theta^{\beta x(fd+1)} \leftarrow \varnothing$
**for** $\forall a \in A$ **do**
    $\Pi_{fd+1}^{(fd+1)} \leftarrow r(a)$
    **for** $C^{mxk} \in T_a$ **do**
        $F^{fd} \leftarrow FEC(C^{mxk})$
        $\Pi_i^{(fd+1)} \leftarrow F_i^{fd}, \forall 1 \leq i \leq fd$
        $\Theta^{\beta x(fd+1)} \leftarrow \Theta^{\beta x(fd+1)} \cup \Pi^{(fd+1)}$
    **end for**
**end for**
$\Omega_\Theta \leftarrow cls\_training(\Theta^{\beta x(fd+1)})$
```

Algorithm 9 $prediction\_compared(C^{mxk}, W)$

```
$\Phi^{\gamma xk} \leftarrow \varnothing, \gamma \leftarrow 0, L^{\tau x4} \leftarrow \varnothing$
**for** $\forall 1 \leq i \leq m$ **do**
    $\Phi^{\gamma xk} \leftarrow \Phi^{\gamma xk} \cup C^{mxk}(i,:)$
    **if** $(\gamma == W) \vee (i == m)$ **then**
        $F^{fd} \leftarrow FEC(\Phi^{\gamma xk})$
        $B_1^4 \leftarrow i - W + 1, B_2^4 \leftarrow i, B_4^4 \leftarrow W$
        $B_3^4 \leftarrow cls\_prediction(\Omega_\Theta, F^{fd})$
        $L^{\tau x4} \leftarrow L^{\tau x4} \cup B^4$
        $\Phi^{\gamma xk} \leftarrow \varnothing, \gamma \leftarrow 0$
    **end if**
**end for**
$return\ L^{\tau x4}$
```

1) Determining degree of polynomial: Degree of polynomial d is an input for the training and prediction stages for NSW. We experimented with the degrees 1-9 and observed that the optimal value for d is 8. Degrees which are higher than 9 are not included in our tests to avoid increasing computational complexity further though there is a chance for a degree higher than 9 could perform better. The results belonging to the degrees outperformed by degree 8 are given in Table XI. In Table XI, *D*, *A* and *T* stand for degree, accuracy (%) and TPR respectively.

D	**A**	**T**
6 5 9 7	11 33 78 89	1
1-4	11	0

(a) T1-P1

D	**A**	**T**
1-4, 6-7 5, 9	22 56	0

(b) T2-P2

D	**A**	**T**
1-6 9	22 78	0
7	89	1

(c) T1-P2

D	**A**	**T**
1-7 9	11 33	1

(d) T2-P1

TABLE XI: Degrees outperformed by optimal degree

The number of the samples which should exist in an accumulating window is $d+1$ for degree d due to general properties of polynomials. Hence, for degree d, the number of samples existing in a training instance, i.e. m for a $C^{mxk} \in T_a$ should take $d+1$ as its minimum. Increasing d allows our method to process larger training instances. However, keeping the size of an individual training instance as small as possible reduces computational complexity in addition to the performance gain obtained by reducing the degree. In this regard, d being 9 indicates that the minimum number of samples existing in a training instance should be 10, which marks the point where we stop experimenting with a higher d.