# TPM Feature Set: a Universal Algorithm for Spatial-Temporal Pressure Mapping Imagery Data

Bo Zhou, Paul Lukowicz

German Research Center for Artificial Intelligence and
Technical Universiy of Kaiserslautern
bo.zhou@dfki.de, paul.lukowicz@dfki.de

***Abstract*—There have been many studies in recent years using the Textile planar Pressure Mapping (TPM) technology for computer-human interactions and ubiquitous activity recognition. A TPM sensing system generates a time sequence of spatial pressure imagery. We propose a novel, comprehensive and unified feature set to evaluate TPM data from the space and time domain. The initial version of the TPM feature set presented in this paper includes 663 temporal features and 80 spatial features. We evaluated the feature set on 3 datasets from past studies in the scopes of ambient, smart object and wearable sensing. The TPM feature set has shown superior recognition accuracy compared with the ad-hoc algorithms from the corresponding studies. Furthermore, we have demonstrated the general approach to further reduce and optimise the feature calculation process for specific applications with neighbourhood component analysis.**

***Index Terms*—textile pressure mapping; data analysis; machine learning algorithm.**

## I. Introduction

Textile Pressure Mapping (TPM) is an emerging technology for ubiquitous and wearable activity recognition. TPM technology measures the distribution of the planar pressure force on a surface, which is omnipresent during all sorts of physical interactions and activities. Dementyev et al. [1] used a wrist-worn Force Sensitive Resistor (FSR) array to detect hand gestures. Cheng et al. [2] proposed a system to detect tongue control gestures with a face-worn TPM patch. Pressure mat placed on the chair surfaces to detect seating postures have also been studied in [3] [4]. Sundholm et al. [5] have demonstrated that sports exercises can be recognized from a sports mat which sense the pressure distribution. Schneegass et al. [6] investigated using a pressure matrix as a sleeve for the forearm to recognize writing gestures.

Many of the researches mentioned above are ad hoc designed on the hardware, software, and algorithm levels. Since each of these studies encloses many aspects, including the hardware, software and activity recognition, none of them are focused on discussions in the algorithm.

Efforts have been devoted to bringing forward a unified solution to push the pressure mapping technology forward in the field of ubiquitous computing and the internet of things. A general hardware architecture to implement TPM sensing systems in [7] . In [8], a framework is proposed to as a unified solution to help developers who are new to the technology to build and evaluate TPM-enabled activity recognition studies.

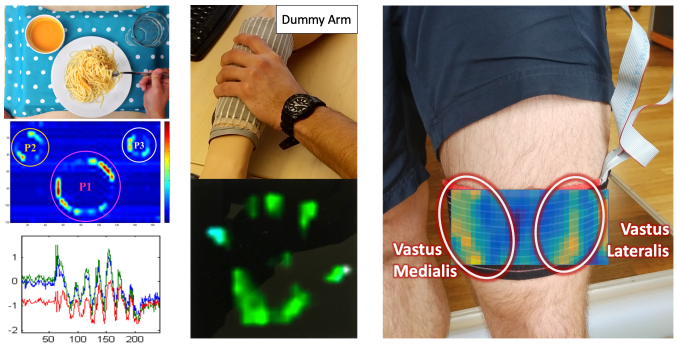



Figure 1. Illustration of the three datasets used in the evaluation: (1) *tablecloth study* [11], (2) *robot skin study* [12], and (3) *leg band study* [13].

However, there lacks a comprehensive investigation into the algorithms of processing the TPM data, especially which features are contributing to the classification results.

TPM imagery has a spatial-temporal data format. Some computer vision techniques for video processing has been applied, such as the work in [9], where neural networks trained for video classification is used to recognize identity from footprint on a TPM carpet. However, TPM imagery is cleaner than camera images in terms of the background scene or objects, obstruction of view, etc. The neural networks also require substantial computational power and are not easily explainable, i.e. which feature or part of the network contributes more to differentiate different activities. Thus many computer vision techniques can be considered over-engineering for processing the TPM imagery. Maximilian et al. [10] proposed a generic feature extraction method on time series. To the best of our knowledge, there lacks a comprehensive and explainable feature analysis scheme that is suitable for the spatial-temporal TPM imagery.

In this paper, we propose the TPM feature set, which comprehensively analyses the TPM data through the time and space domains. The TPM feature set is evaluated with datasets from three empirical studies with different application scenarios. We also investigate which features are contributing more positively to the classification through neighbourhood component analysis. The streamlined methods proposed in this paper can be used to analyse and optimise new datasets from future TPM studies.

In Section II, the general format of the pressure mapping data and terminologies are introduced. Section III explains the detailed generic method, the TPM feature set, to extract information from space and time domains. In Section IV, the spatial and temporal domains are discussed in depth to investigate which features are more relevant with established empirical study datasets. Section V concludes the paper.

## II. TPM Data in the Spatial-Temporal Domains

TPM sensors generate a multi-channel, spatial-temporal data format, which describes the localization of the pressure distribution along the time axis.

Every sensing point is defined as a *pixel*:

$$p(x, y, t) \tag{1}$$

where $x$ and $y$ are the coordinates in the spatial dimensions, and $t$ is the specific time.

At any time $t$, the entire mapping $M$ of the sensor is defined as a *Frame*:

$$F(t) = \{p(x, y, t) \mid (x, y) \in \{M\}\} \tag{2}$$

A temporal sequence from a time window $T$ of *Frames* is defined as a data *Stream*:

$$S_T = \{F(t) \mid t \in \{T\}\} \tag{3}$$

Individual sensing points have limited information about the activity; therefore, some descriptive values of the frame at a given time $t$ are calculated as *Frame Descriptors*:

$$des_i(t) = Func_i(F(t)) \tag{4}$$

Another approach to abstract the *stream* from a time window is to perform per-pixel operations along the time axis, resulting in individual frames that represent the stream. We call these frames as *Key Frames*:

$$KF_i(t) = Func_i(S_T)) \tag{5}$$

## III. the TPM Feature Set

Figure 2 shows the general workflow of calculating the TPM feature set from the space and time domains. Temporal features are extracted from sequences of simple frame descriptors $des_i(t)$. Spatial features are calculated from 2-dimensional key frames $KF_i(t)$. The initial version of the TPM feature set includes 663 ($17 \times 39$) temporal features and 80 ($8 \times 10$) spatial features.

### A. Frame Descriptors

Treat $F(t)$ as a set, the TPM feature set calculates the following $des_i(t)$:

- average value

$$des_1(t) = mean(F(t)) = \frac{1}{|M|} \sum_{(x,y)}^{\{M\}} p(x, y, t) \tag{6}$$

- variance

$$des_2(t) = \frac{1}{|M|} \sum_{(x,y)}^{\{M\}} (p(x, y, t) - mean(F(t)))^2 \tag{7}$$

- range

$$des_3(t) = p_{MAX}(t) - p_{MIN}(t) \tag{8}$$

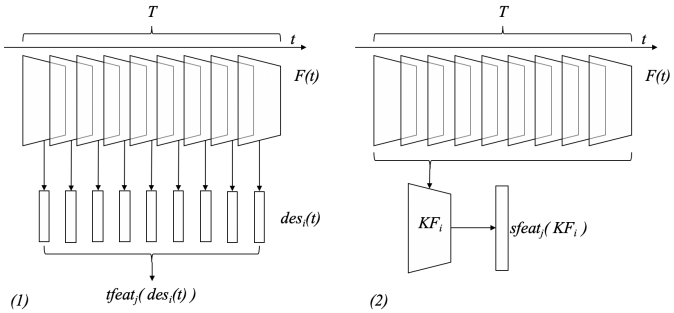



Figure 2. (1) Temporal and (2) spacial feature extraction process.

- entropy

$$des_4(t) = -\sum_{(x,y)}^{\{M\}} p(x, y, t) \cdot log_2\, p(x, y, t) \tag{9}$$

- mean absolute deviation

$$des_5(t) = \frac{1}{|M|} \sum_{(x,y)}^{\{M\}} (p(x, y, t) - mean(F(t))) \tag{10}$$

- the center of mass (CoM) coordinate x and y (weighted by pixel value) $des_6(t)$ and $des_7(t)$
- the centroid coordinate (unweighted, only considering the contour after filtering the frame with a threshold). $des_8(t)$ and $des_9(t)$. Here the threshold is defined as

$$mean(F(t)) - 0.25 \cdot (mean(F(t)) - p_{min}(t)) \tag{11}$$

- area (the count of pixels that are above the threshold) $des_{10}(t)$
- $des_{11}(t)$ to $des_{17}(t)$ Hu's seven image moments [14]

For a matrix of binary values, the CoM is identical to the centroid; but for a matrix with multi-values that describes a profile, the CoM shows how the pixel value density is focused while the centroid shows only the geometric center. $des_1(t)$, $des_6(t)$ and $des_7(t)$ are mathematically identical to the first three central moments in the literature on image moments.

### B. Temporal Features

Any sequence of frame descriptors is denoted as $des_i(t) \in \{des_1(t), des_2(t), ...\}$. Then, from the temporal sequence within a window of length $T$ (sliding window or spotted event), temporal features can be calculated:

- average

$$tfeat_1 = \frac{1}{|T|} \sum_{t}^{\{T\}} des_i(t) \tag{12}$$

- variance

$$tfeat_2 = \frac{1}{|T|} \sum_{t}^{\{T\}} (des_i(t) - tfeat_1)^2 \tag{13}$$

- range

$$tfeat_3 = des_{i_{MAX}} - des_{i_{MIN}} \tag{14}$$

- skewness, that describes the asymmetry of the data

$$tfeat_4 = \frac{\frac{1}{|T|} \sum_t^T (des_i(t) - tfeat_1)^3}{\left(\frac{1}{|T|} \sum_t^T (des_i(t) - tfeat_1)^2\right)^{3/2}} \tag{15}$$

- kurtosis, that measures how outlier-prone the temporal sequence's distribution is

$$tfeat_5 = \frac{\frac{1}{|T|}\sum_t^T (des_i(t) - tfeat_1)^4}{\left(\frac{1}{|T|}\sum_t^T (des_x(t) - tfeat_1)^2\right)^2} \quad (16)$$

- waveform length [15]

$$tfeat_6 = \sum_t^T -1(des_i(t+1) - des_i(t)) \quad (17)$$

- sum of values greater than mean

$$tfeat_7 = \sum_t^T (des_i(t) \mid des_i(t) > tfeat_1) \quad (18)$$

- the power spectrum density of $des_i$ is calculated with fast Fourier transform as $PSD(n)$, $n \in N$ is the frequency in the spectrum. Following features are calculated from $PSD(n)$: average magnitude

$$tfeat_8 = \frac{1}{N}\sum_n^N PSD(n) \quad (19)$$

- mean frequency

$$tfeat_9 = \frac{\sum_n^N n \cdot PSD(n)}{\sum_n^N PSD(n)} \quad (20)$$

- $N$ is divided to 5 equal frequency bands, the average values of each band is $tfeat_{10}$, $tfeat_{11}$, $tfeat_{12}$, $tfeat_{13}$, $tfeat_{14}$.
- A wavelet transform scalogram is calculated with the LTFAT toolbox [16], with $J = 4$ filterbank iterations. The coefficient vector of each filterbank is $C(j), j \in [0, 4]$.
- $tfeat_{15}$, $tfeat_{20}$, $tfeat_{25}$, $tfeat_{30}$, $tfeat_{35}$ are the mean value of each coefficient vector;
- $tfeat_{16}$, $tfeat_{21}$, $tfeat_{26}$, $tfeat_{31}$, $tfeat_{36}$ are the variance of each coefficient vector;
- $tfeat_{17}$, $tfeat_{22}$, $tfeat_{27}$, $tfeat_{32}$, $tfeat_{37}$ are the range of each coefficient vector;
- $tfeat_{18}$, $tfeat_{23}$, $tfeat_{28}$, $tfeat_{33}$, $tfeat_{38}$ are the skewness of each coefficient vector;
- $tfeat_{19}$, $tfeat_{24}$, $tfeat_{29}$, $tfeat_{34}$, $tfeat_{39}$ are the kurtosis of each coefficient vector;

### C. Key Frames

From a time window, a *key frame* can be one particular frame that has special frame descriptor values, such as the maximum or minimum of $des_i(t)$. A *key frame* can also be calculated from the stream of the window through pixel-wise operations. 8 key frames are calculated in the TPM feature set:

- per pixel average of all frames

$$KF_1 = \frac{1}{|T|}\sum_t^{\{T\}} F(t) \quad (21)$$

- sum of per pixel differences

$$KF_2 = \sum_t^{\{T-1\}} (F(t+1) - F(t)) \quad (22)$$

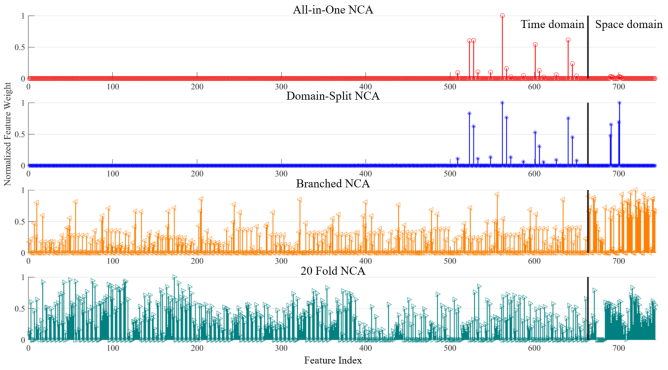



Figure 3. Feature weight distributions of different NCA division methods (tablecloth dataset)

- sum of only the positive or negative values of per pixel differences

$$KF_3 = \mid \sum_t^{\{T-1\}} ((F(t+1) - F(t)) > 0) \mid \quad (23)$$

$$KF_4 = \mid \sum_t^{\{T-1\}} ((F(t+1) - F(t)) < 0) \mid \quad (24)$$

- the frame which has the maximum mean pixel value as $KF_5$ and the frame with the minimum mean value as $KF_6$
- the frame with the maximum standard deviation from the stream as $KF_7$
- the per pixel average of the frames, whose pixel value is greater than the frame pixel average

$$KF_8 = \frac{1}{|T|}\sum_t^{\{T\}} (F_p(t)) \quad (25)$$

$$F_p(t) = \begin{cases} p(x,y,t) & \text{if } p(x,y,t) \geq mean(F(t)) \\ 0 & \text{if } p(x,y,t) < mean(F(t)) \end{cases}$$

### D. Spatial Features

Various image processing techniques can then be used to extract information from those key frames. Image moments are proven to be helpful shape descriptors as spatial features $sfeat_j(KF_i)$ through previous studies. In the TPM feature set, we use the 3 central moments and Hu's 7 invariant moments [14], which are rotation, translation and scale invariant.

## IV. Evaluation and Feature Selection

In this section, we evaluate how different combinations of frame descriptors - temporal feature pairs, and key frame - spatial feature pairs contribute to the classification accuracy. The datasets used are from various past studies in different setting scenarios.

### A. General Approach

The evaluation process can be divided into four parts:

*1) Convert the data stream into features:* From the time domain, first, temporal sequences of the 17 frame descriptors $des_i(t)$ are calculated from every stream. Within every $des_i(t)$, a sliding window is performed. Every window is denoted as $n \in N$. The data in the window is multiplied with a Tukey window with $r = 0.2$, to bring the start and end of the window to zero. For every sliding window, 39 temporal features $tfeat_j(des_i), j \in 1, 2, ...39, i \in 1, 2, ...17$ are calculated. In the spatial domain, first the input data stream is cropped with the same window size and window step as the sliding window for $des_i(t)$, but the outputs are the smaller length of streams, and no Tuken window is applied. Within each window of streams, 8 key frames $KF_i$ are calculated. Overall 10 spacial features is calculated from every key frame $sfeat_j(KF_i), j \in 1, 2, ...10, i \in 1, 2, ...8$.

*2) Baseline cross-validation:* To carry out balanced training, all classes are trimmed to the same amount of windows by random selecting. The amount of windows is the class that has the least windows. K-fold cross-validation is performed with multiple classifiers, and the accuracy is used to compare different classifier's results.

*3) Feature selection:* The feature weight evaluation is performed using neighbourhood component analysis (NCA) [17]. The method ranks the most relevant features that contribute to the classification. Since the features are calculated from two levels of information: temporal features are calculated first by reducing the space domain to frame descriptors, then to the time domain features; as spatial features are calculated first by reducing the time domain to key frames. Thus, the feature weight result can either be presented as a *feature weight vector* or as a *feature weight matrix* for either the temporal or spacial feature methods.

*4) Feature reduction:* The top-weighted features are selected to perform the same cross-validation. For comparison, the least weighted features are also evaluated separately.

Principle Component Analysis (PCA) [18] is another commonly used technique for reducing feature dimensions. The method removes redundancy and outputs a set of eigenvectors that best describes the variance of the dataset. Each component is orthogonal to the preceding one so that the eigenvectors are uncorrelated and thus without redundancy. However, PCA itself does not take the class label information, it only analyses the data distribution to remove redundancy but not irrelevant features. Typically, PCA is used as a step after calculating the features, and before feeding the information to classifiers. Therefore, we use NCA instead of PCA to find the features that are more contributive to distinguishing different classes.

### B. Datasets

Three past studies are taken for comparison, they are codenamed as: *tablecloth* [11], *robot skin* [12], and *leg band* [13] as shown in Figure 2.

In the tablecloth study [11], a TPM fabric with a 30-by-42 matrix is placed on a dining tablet to detect dining related actions. A main dish plate, a salad bowl, and a glass are placed on it. Participants eat various food of different textures, that would require different actions for dining the food with a knife and a fork. The force of the actions can propagate through the cutlery and plates to the tablecloth surface. The 7 action classes are: stir, scoop, cut, poke, scoop, collect and replace. The sliding window is chosen with 2 second period and 1 second window step. 10 participants each took part in 8 recordings.

In the robot skin study [12], a TPM fabric with a 20-by-20 matrix is used to detect 7 emotionally related touch gestures onto a dummy arm or a surface, including grab, poke, press, push, scratch, pinch and stroke. The gestures are already segmented based on matrix activation, since when there is no gesture, the matrix is not being pressed. In total, 24 participants took part in 2 recordings. Each recording includes 16 repetitions of every gesture.

In the leg band study [13], a TPM fabric with an 8-by-16 matrix is embedded in an elastic compression band that is placed on the thigh as users take part in gym leg exercises. The sensor detects the surface pressure of the leg muscles as planar pressure mechanomyography. The 5 activity classes are: working out with a cross trainer, leg press, seated leg curl and leg extension, plus a class contains all non-workout activities. Based on the activity's characteristic, the sliding window is chosen as 4 seconds wide, the window step is 20% of the window size. Six participants have recorded 4 sessions each.

In this paper, all the participants' data are merged together as one dataset for every study (person independent - inclusive case). Every sliding window or gesture is one sample. The tablecloth dataset has 10815 samples, robot skin 5376 samples, and leg band 28425 samples.

### C. Neighborhood Component Analysis (NCA)

This subsection briefly explains the NCA method published in [17] and implemented in Matlab as `fscnca`. The NCA method assumes a feature weight vector $w$ as a variable to be multiplied with the features, and uses an approximate solver to find the optimal weight vector that maximizes the correct classification probability (the objective function). (The mathematical symbols for NCA explanation are not related to the rest of this paper for the TPM feature set.)

For a d-dimensional dataset of $N$ training points, all the points from the dataset are taken as a query point once $x_i$. For each query point, the other points can be taken as its reference point as a probability $p_{ij}$ derived from their weighted distance enclosed in a kernel function. The probability that this query point $x_i$ is correctly classified is defined as the probability summation of the reference points that has the same class, similar to a K-nearest neighbour classifier.

The objective function is the average of all the points' correct classification probability. After unfolding the relationship, the objective function can be written as a differentiable function of the feature weight vector, with a tunable parameter $\lambda$ which is multiplied with the weight vector's term in the objective function:

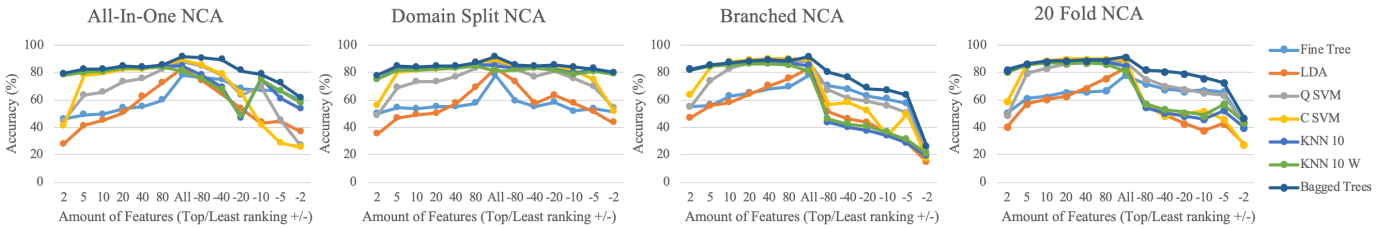



Figure 4. Accuracy with varying amount of selected features comparison of four NCA division methods (tablecloth dataset)

$$F(w) = \sum_{i=1}^{N} \left( \sum_{j=1, j \neq i}^{N} P_{ij} y_{ij} - \lambda \sum_{l=1}^{d} w_l^2 \right) \tag{26}$$

where $y_{ij} = 1$ if the query point and the reference point has the same class. Since $F(w)$ is differentiable, its maxima can be approximated with algorithms, such as stochastic gradient descent (SGD) [19], to find out the feature weight vector $w$ that maximizes the objective function $F(w)$. In other words, NCA finds the best feature weight combination that yields the highest correct classification probability.

### D. NCA division approaches on high dimensional features

A problem of NCA is that when most of the features contribute to the classification, the approximation may return to only very few highly weighted features while the others remain close to zero weight. This leaves the classification result relatively low with selected high weighted features. Our solution is to segment these features and perform NCA on smaller batches, then combine the feature weights. In this work, four NCA division approaches are investigated:

- All-in-One: all the features are taken under NCA as once.
- Space-time domain split: features are split into two groups: spatial domain features and temporal domain features.
- Branched: features are more detailed separated into branches. In the time domain, a branch is all the temporal features from one frame descriptor; in the space domain, a branch is all the spatial features from one key frame.
- K-fold: all features are randomized and split into K equal partitions. One NCA is performed for each partition.

In the segmented feature groups, the resulting feature weight vectors are normalized within each group before being concatenated into one vector. The results of the four different feature division approaches on the smart tablecloth data are shown in Figure 3. From the result, All-in-One and domain-split NCA return similar weight for the time domain features, with zero weight for most of the features. The domain split NCA gives higher weight on the spatial features as a result of normalization before merging, but the feature indexes that are higher than approximate zero are the same between the two approaches. In the branched and 20-fold NCA, however, many more features are given higher weight.

To compare which approach is better, cross-validation with the highest ranking features, in comparison with the lowest ranking features can be used. A better approach should meet the following criteria:

- Higher accuracy with the same number of top ranking features compared to other approaches.
- Greater difference between highest accuracy and the accuracy with the least ranking features, than the difference between highest accuracy and the accuracy with the top ranking features.
- With the same amount of features, top ranking features should in general result in higher accuracy than least ranking features.

### E. NCA Evaluation

To evaluate which approach yields better feature selection, cross-validation from the top or least ranking features are performed. For performance reasons, top or least 2, 5, 10, 20, 40 and 80 features are chosen. The NCA algorithm should have greater influence on the KNN classifiers since the basic principle is similar (Euclidean distance to the training data samples). In this evaluation, a variety of classifiers are chosen:

1) classification tree with 100 maximum splits and Gini's diversity index split criterion (Fine Tree)
2) linear discriminant analysis (LDA)
3) support vector machine with a quadratic kernel (Q SVM)
4) support vector machine with a cubic kernel (C SVM)
5) K-nearest neighbor with equally weighted Euclidean distance and K=10 (KNN 10)
6) K-nearest neighbor with squared inversely weighted Euclidean distance and K=10 (KNN 10 W)
7) Ensemble of 30 decision tree learners (Bagged Trees)

The results are shown in Figure 4.

For many classifiers, all-in-one and domain split NCA has a near symmetric accuracy distribution centered at all features; sometimes with the least ranking features, there are higher accuracy points than the corresponding top ranking features. From this, we concluded that the feature weights derived by these two methods are no better than random selection. Branched and 20 Fold NCA, on the other hand, in general, meet the criteria listed above, and have a similar trend of the accuracy values. The highest ranking features result in higher accuracy values than the lowest ranking features.

The top 2 ranking features already result in over 80% accuracy for Bagged Trees and the two KNN classifiers. While for the other classifiers, Fine Tree, LDA and SVM, the accuracy values are significantly lower. This may because these classifiers work by separating the feature space with modelled boundaries, while KNN and bagged trees do not use such boundaries to distinguish different classes. The data's

nature may not fit very well with the classifiers' algorithms, e.g., the data may not have clean-shaped boundaries, or the same class may have several clusters. However, this cannot be further investigated at this point due to the high dimensionality.

The least 2 ranking features result in close to chance level (14.3% for 7 classes) accuracy values, thus means the NCA successfully identify the less relevant features. As the number of features taken grows, the accuracy of both top and least ranking features increase, but the top ranking features give higher accuracy than the least ranking ones.

As branched NCA is not a generic approach, and K-Fold NCA can be performed on any feature sets, this work will continue with K-Fold NCA. Figure 5 shows the top ranking features but with more amount of taken features until all of them are chosen. From it, the accuracy has come to a stable level close to 90% between 10 to 160 features for most classifiers except for LDA and Fine Tree; while from 240 features on, the accuracy has another increase that is on the similar level with all the features. This shows that only the top 10 features are sufficient for this dataset for moderately high accuracy, and 240 features are adequate to explain all the class discriminant as good as with all features.

### *F. Application Variance and Discussions*

To be displayed only as a linear vector of values as in Figure 3 is not sufficient to tell which feature calculation method is more relevant. Therefore, the feature weight vector is reshaped into two 2-dimensional matrices according to the frame descriptor - temporal feature combination or key frame - spacial feature combination as a feature weight matrix ($FWM$). For the tablecloth dataset, the temporal feature weight matrix $FWM_t$ is shown in Figure 6, and the spacial feature weight matrix $FWM_s$ is in Figure 9(1). From $FWM_t$, it can be seen that some temporal features have no contribution, such as skewness, kurtosis, including the skewness and kurtosis for the wavelet transform. Some frame descriptors are more important, such as $des_2$ variance, $des_3$ range, $des_5$ mean absolute deviation. All the 7 Hu's moments $des_{11}$ to $des_{17}$ are less important. It is possibly a result that in this dataset, the objects are all plates or glasses, and their footprints are all circular. Hence, the shape descriptors are not contributing to the activity. From $FWM_s$, the key frames describe the static values, such as $KF_1$ and $KF_2$ are less contributive, while the key frames that describe the dynamic changes all have greater feature weights.

Two other datasets are evaluated with the same process, and the resulting plots of 'number of features' - accuracy plots are in Figure 10. (SVM classifiers are not used for evaluation here due to performance constraints.) Referring to the NCA evaluation criteria, NCA has effectively located relevant features in all datasets. Feature weight matrix are shown in Figure 7, Figure 8, and Figure 9(2)(3). Table I compares the accuracy of the original studies with the TPM feature set. The top 20 features from each dataset are further listed in Table II.

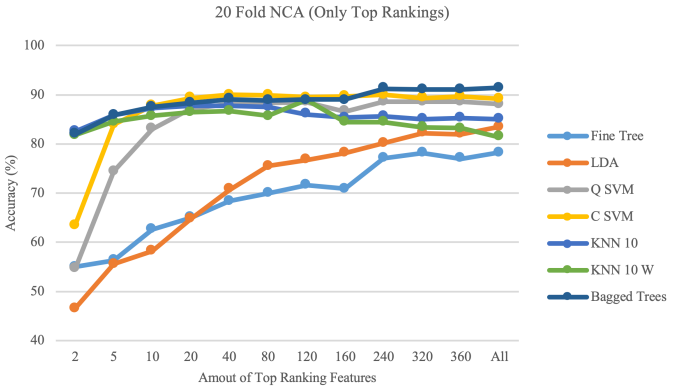



Figure 5. Top ranking features of the 20 fold NCA on the tablecloth dataset to locate the optimal amount of features.

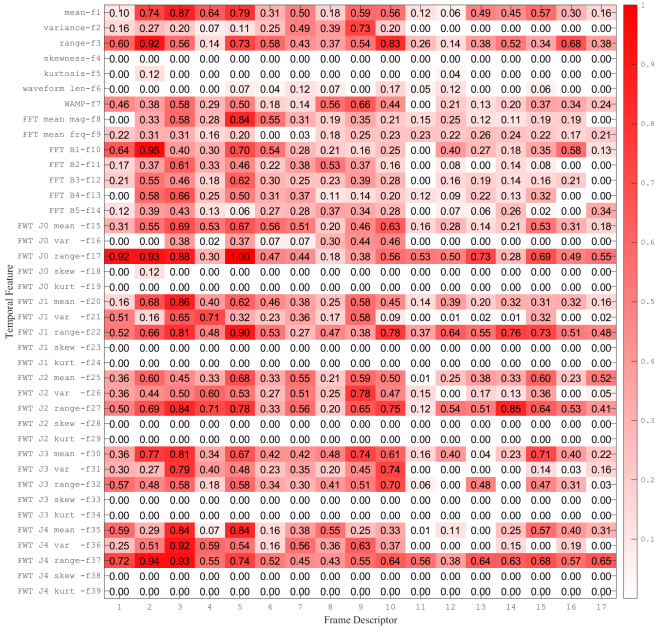



Figure 6. Temporal feature weight - Smart tablecloth dataset.

Comparing the $FWM_t$ and $FWM_s$ of all datasets, highest ranking features are different for specific applications. There are only three features that are present in all datasets' top 40 features. For example, skewness and kurtosis have relatively high weight for the robot skin dataset, and also have higher weights in some of the frame descriptors for the leg band dataset. The FFT features have almost no weight in the robot skin dataset, while these features are significantly relevant in the tablecloth and leg band dataset. And when FFT features have more weight, wavelet transform features also have more

TABLE I. ACCURACY COMPARISON OF THE ORIGINAL STUDIES AND THE TPM FEATURE SET

| Dataset | Original Study | TPM Feature Set |
|---|---|---|
| Tablecloth | 91.2% | 91.4 % |
| Robot Skin | 92.7% | 94.7 % |
| Leg Band | 81.7 % | 98.2 % |

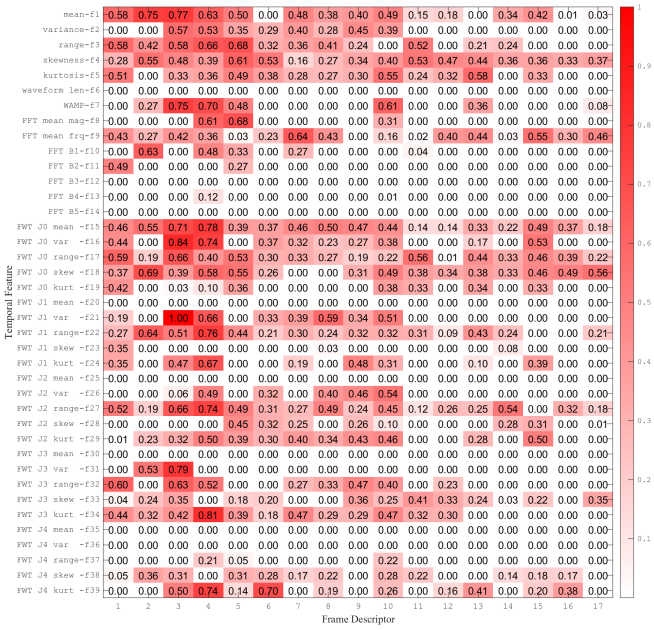



Figure 7. Temporal feature weight (robot skin dataset)

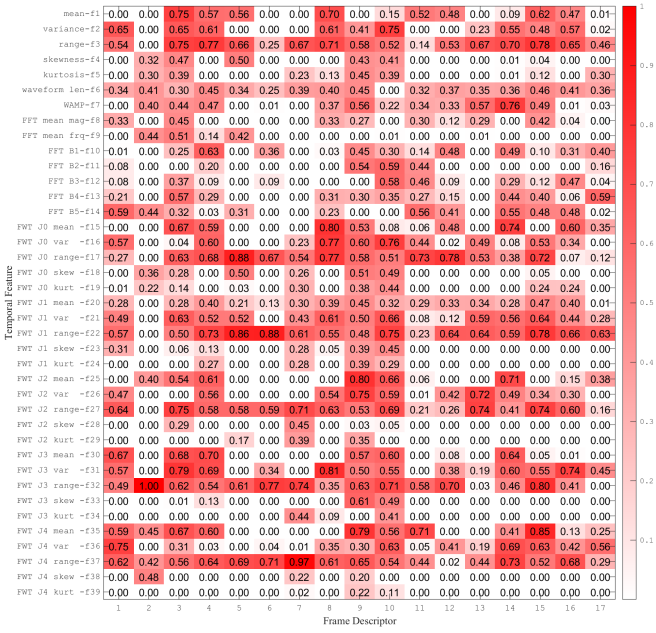



Figure 8. Spatial feature weight (leg band dataset)

weight. This is expected since both FFT and wavelet transform describes frequency information. In the leg band dataset, Hu's 7 moments as frame descriptors have significantly higher weight than the other two dataset. Spatial features on average have less weight in the tablecloth and robot skin datasets than in the leg band datasets.

The key conclusion is that, for different applications, the TPM sensor data exhibit different natures.

### G. Performance Benchmark

We evaluated the computational performance with a dataset recording file (.mat format) of 286MB (Leg Band dataset

TABLE II. TOP RANKING FEATURES

**Tablecloth dataset**

| **Ranking** | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| Des/KF | 5 | 2 | 2 | 3 | 2 | 1 | 3 | 2 | 5 | 3 |
| Feature | 17 | 10 | 37 | 37 | 17 | 17 | 36 | 3 | 22 | 17 |
| Domain | T | T | T | T | T | T | T | T | T | T |
| Ranking | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| Des/KF | 3 | 3 | 14 | 5 | 5 | 3 | 3 | 6 | 10 | 3 |
| Feature | 1 | 20 | 27 | 35 | 8 | 35 | 27 | 3 | 3 | 22 |
| Domain | T | T | T | T | T | T | T | **S** | T | T |

**Robot Skin dataset**

| Ranking | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| Des/KF | 3 | 3 | 4 | 4 | 4 | 3 | 4 | 6 | 3 | 4 |
| Feature | 21 | 16 | 34 | 8 | 1 | 31 | 15 | 7 | 1 | 22 |
| Domain | T | T | T | **S** | **S** | T | T | **S** | T | T |
| Ranking | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| Des/KF | 3 | 2 | 4 | 4 | 4 | 3 | 4 | 3 | 3 | 4 |
| Feature | 7 | 1 | 27 | 39 | 16 | 1 | 7 | 6 | 15 | 2 |
| Domain | T | T | T | T | T | **S** | **S** | **S** | T | **S** |

**Leg Band dataset**

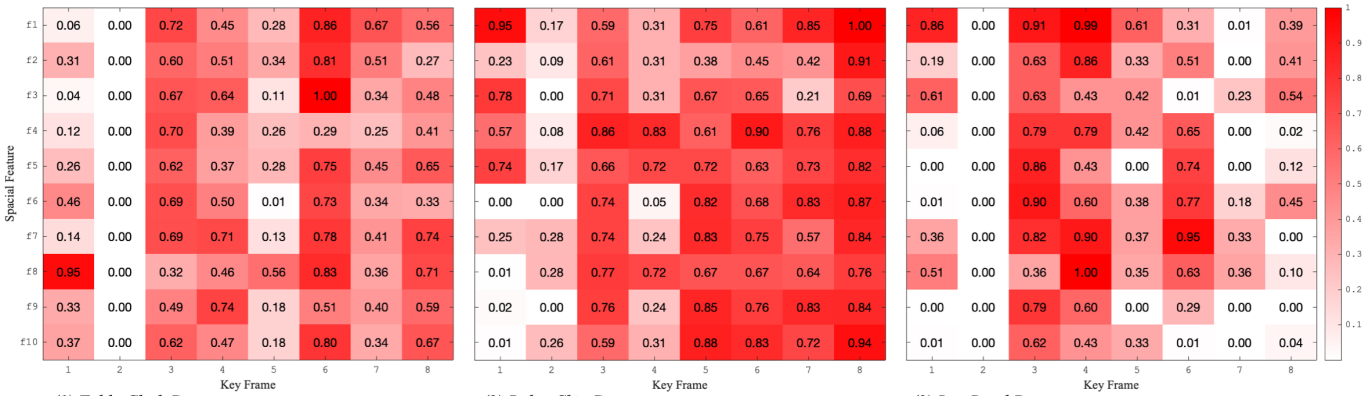
| Ranking | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| Des/KF | 2 | 7 | 6 | 5 | 5 | 15 | 8 | 15 | 9 | 8 |
| Feature | 32 | 37 | 22 | 17 | 22 | 35 | 31 | 32 | 25 | 1 |
| Domain | T | T | T | T | T | T | T | T | T | **S** |
| Ranking | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| Des/KF | 8 | 9 | 3 | 15 | 12 | 15 | 8 | 4 | 8 | 6 |
| Feature | 15 | 35 | 31 | 3 | 17 | 22 | 16 | 3 | 17 | 32 |
| Domain | T | T | T | T | T | T | T | T | T | T |

person 1 recording 1). The benchmark was carried out on a 2018 MacBook Pro with a six-core 2.6GHz Intel Core i7 processor, and Matlab 2019a. The total frame descriptors calculation took 43.87s and total key frames 2.47s. All the temporal features from all frame descriptors took 350.05s and the spatial features 0.765s. During the temporal feature calculation, the most time consuming process is the fast wavelet transform, which takes 281.35s out of the 350.05s. The 20 fold NCA with all the recordings from the leg band dataset took 926.83s.

However, since the feature set is meant to help explore the useful features for specific data set offline, the computational requirement is less important. With the NCA optimization method, developers can further reduce and select the features to be computed based on their specific requirements.

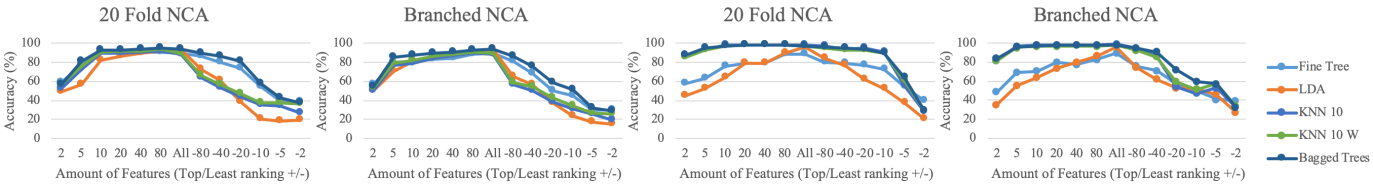
## V. Conclusion

A generic feature calculation method, the TPM feature set, is proposed in this paper. Built upon various relevant studies, it can be used to extract information from both the space and time domains for the TPM imagery. Through our evaluation, our approach shows superior accuracy compared to the original studies in which the datasets were published with ad hoc algorithms. Not all features contribute equally, and the




*(1) Table Cloth Dataset* *(2) Robot Skin Dataset* *(3) Leg Band Dataset*

Figure 9. Spatial feature weight matrices of the three datasets.




*(1) Robot Skin dataset* *(2) Leg Band dataset*

Figure 10. Feature number against accuracy for the Robot Skin and Leg Band dataset with 20 fold and branched NCA.

feature weights vary with different applications. Neighborhood component analysis can be used to locate and explain the more contributing features and further optimise a system by reducing feature calculation efforts.

## References


[1] A. Dementyev and J. A. Paradiso, "Wristflex: low-power gesture input with wrist-worn pressure sensors," in Proceedings of the 27th annual ACM symposium on User interface software and technology. ACM, 2014, pp. 161–166.

[2] J. Cheng et al., "On the tip of my tongue: a non-invasive pressure-based tongue interface," in Proceedings of the 5th Augmented Human International Conference. ACM, 2014, p. 12.

[3] S. Mota and R. W. Picard, "Automated posture analysis for detecting learner's interest level," in 2003 Conference on Computer Vision and Pattern Recognition Workshop, vol. 5. IEEE, 2003, pp. 49–49.

[4] B. Zhou et al., "Smart blanket: A real-time user posture sensing approach for ergonomic designs," in International Conference on Applied Human Factors and Ergonomics. Springer, 2017, pp. 193–204.

[5] M. Sundholm, J. Cheng, B. Zhou, A. Sethi, and P. Lukowicz, "Smart-mat: Recognizing and counting gym exercises with low-cost resistive pressure sensing matrix," in Proceedings of the 2014 ACM international joint conference on pervasive and ubiquitous computing. ACM, 2014, pp. 373–382.

[6] S. Schneegass and A. Voit, "Gesturesleeve: using touch sensitive fabrics for gestural input on the forearm for controlling smartwatches," in Proceedings of the 2016 ACM International Symposium on Wearable Computers. ACM, 2016, pp. 108–115.

[7] B. Zhou, J. Cheng, M. Sundholm, and P. Lukowicz, "From smart clothing to smart table cloth: Design and implementation of a large scale, textile pressure matrix sensor," in International Conference on Architecture of Computing Systems. Springer, 2014, pp. 159–170.

[8] B. Zhou et al., "Tpm framework: a comprehensive kit for exploring applications with textile pressure mapping matrix," in UBICOMM 2017 : The Eleventh International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies. IARIA, 2017.

[9] M. S. Singh et al., "Transforming sensor data to the image domain for deep learning—an application to footstep detection," in 2017 International Joint Conference on Neural Networks (IJCNN). IEEE, 2017, pp. 2665–2672.

[10] M. Christ et al., "Time series feature extraction on basis of scalable hypothesis tests (tsfresh–a python package)," Neurocomputing, vol. 307, 2018, pp. 72–77.

[11] B. Zhou, J. Cheng, P. Lukowicz, A. Reiss, and O. Amft, "Monitoring dietary behavior with a smart dining tray," IEEE Pervasive Computing, vol. 14, no. 4, 2015, pp. 46–56.

[12] B. Zhou et al., "Textile pressure mapping sensor for emotional touch detection in human-robot interaction," Sensors, vol. 17, no. 11, 2017, p. 2585.

[13] B. Zhou, M. Sundholm, J. Cheng, H. Cruz, and P. Lukowicz, "Measuring muscle activities during gym exercises with textile pressure mapping sensors," Pervasive and Mobile Computing, vol. 38, 2017, pp. 331–345.

[14] M.-K. Hu, "Visual pattern recognition by moment invariants," IRE transactions on information theory, vol. 8, no. 2, 1962, pp. 179–187.

[15] Z. O. Khokhar, Z. G. Xiao, and C. Menon, "Surface emg pattern recognition for real-time control of a wrist exoskeleton," Biomedical engineering online, vol. 9, no. 1, 2010, p. 41.

[16] Z. Průša, P. L. Søndergaard, N. Holighaus, C. Wiesmeyr, and P. Balazs, "The Large Time-Frequency Analysis Toolbox 2.0," in Sound, Music, and Motion, ser. Lecture Notes in Computer Science, M. Aramaki, O. Derrien, R. Kronland-Martinet, and S. Ystad, Eds., 2014, pp. 419–442.

[17] W. Yang, K. Wang, and W. Zuo, "Neighborhood component feature selection for high-dimensional data." JCP, vol. 7, no. 1, 2012, pp. 161–168.

[18] H. Hotelling, "Analysis of a complex of statistical variables into principal components." Journal of educational psychology, vol. 24, no. 6, 1933, p. 417.

[19] E. Moulines and F. R. Bach, "Non-asymptotic analysis of stochastic approximation algorithms for machine learning," in Advances in Neural Information Processing Systems, 2011, pp. 451–459.