Feature Extraction and Classification using Leading Eigenvectors: Applications in Biomedical and Multi-Modal mHealth Data

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

Feature Extraction and Classification using Leading Eigenvectors: Applications to Biomedical and Multi-Modal mHealth Data

GEORGINA COSMA[1,2], (Member, IEEE), MARTIN T. MCGINNITY[1,3], (Member, IEEE)
[1]Department of Computer Science, Loughborough University, UK (e-mail: gcosmaresearch@outlook.com)
[2]Department of Computing at School of Science and Technology, Nottingham Trent University, UK (e-mail: georgina.cosma@ntu.ac.uk)
[3]Intelligent Systems Research Centre, Ulster University, Northern Ireland, Derry, UK (e-mail: martin.mcginnity@ntu.ac.uk)

Corresponding author: Georgina Cosma (e-mail: gcosmaresearch@outlook.com)

This work was supported by The Leverhulme Trust Research Project Grant RPG-2016-252 entitled "Novel Approaches for Constructing Optimised Multimodal Data Spaces".

ABSTRACT Eigendecomposition is the factorisation of a matrix into its canonical form, whereby the matrix is represented in terms of its eigenvalues and eigenvectors. A common and important step is the reduction of the data to a kernel matrix, also known as a Gram matrix which is used for machine learning tasks. A significant drawback of kernel methods is the computational complexity associated with manipulating kernel matrices. This paper proposes an efficient approach to apply eigendecomposition methods for significantly improving the speed and accuracy of adopting Singular Value Decomposition and Nyström approximation methods for classification tasks. Experiments were conducted with 14 biomedical datasets to compare classification performance when taking as input into a classifier matrices containing: 1) leading eigenvectors which result from each approximation method; and 2) matrices which result from constructing the patient-by-patient Gram matrix. Experiments revealed significant differences ($p < 0.05$) in classifier performance in terms of accuracy and time when the two methods were used. In addition, results show that using the proposed approach, Singular Value Decomposition becomes faster and more accurate than Nyström based methods. Furthermore, experiments were carried out using multi-modal mHealth time series datasets for ten subjects of diverse profiles while performing several physical activities. Experimental results using the mHealth data demonstrate the superiority of the proposed approach when using Singular Value Decomposition instead of Nyström methods for feature extraction and classification of large datasets using a Deep Sequential model. The results provide evidence to support the main hypothesis of this paper, that using as input into a classifier the leading eigenvectors significantly improves classifier performance in terms of accuracy and time compared to using Gram matrices. The significance of the proposed approach is that it can make feature extraction methods more accessible on large-scale unimodal and multi-modal data which are becoming common in many applications.

INDEX TERMS biomedical data, classification, machine learning, time-series, human activity recognition, multi-modal data.

I. INTRODUCTION

LOW-rank matrix decompositions are important in the application of kernel methods to large-scale learning problems. High-dimensional data is represented in more than two or three dimensions and it can be difficult to manipulate and interpret. One approach to dealing with high-dimensional data is to assume that the data of interest reside on an embedded non-linear manifold within the higher-dimensional space. If the manifold is of low enough dimensionality, the data can be visualised in a low-dimensional space. Manifold learning is also known as non-linear dimensionality reduction.

Large matrices consist of thousands to millions of matrix entries and performing even simple operations on these matrices becomes a complex task. Feature extraction algorithms are utilised to reduce the data into fewer dimensions and hence to deal with the 'curse of dimensionality', so as to reduce the complexity and improve the efficiency of operating on large matrices by constructing lower-rank matrix approximations of large matrices [1]. Therefore, the task of feature extraction or dimensionality reduction has become common in large-scale applications, including machine learning. Furthermore, the idea behind creating low-rank approximations is that representing the data in a reduced dimensional space removes noise from the data, which then reveals intrinsic structures of the data. For this reason, it is important to create low-rank matrix approximations and utilise these, instead of the full-rank matrices. Many methods have been proposed to construct low-rank approximation of matrices, and these methods rely on the eigenvectors of the kernel matrix. Some of these methods include Independent Component Analysis (ICA) [2], Principal Component Analysis (PCA) (also called Karhunen-Loéve Transform-KLT), Singular Value Decomposition (SVD), Laplacian Eigenmap [3], Multidimensional Scaling (MDS) [4], Spectral clustering [5], Isometric multimanifold learning [6], kernel Fisher linear discriminant analysis [7], and the clustered Nyström method [8]. A common and important step in kernel methods is the reduction of the data to a kernel matrix, also known as a Gram matrix. The Gram matrix is then used for machine learning tasks such as classification, clustering, and dimensionality reduction [9]. A significant drawback of kernel methods is the computational complexity associated with manipulating kernel matrices.

Given a set of n data points, the kernel matrix K is of size $n \times n$, which results in a computational complexity of at least $O(n^2)$ [9]. Furthermore, the majority of kernel methods, such as Singular Value Decomposition, have at their core operations the tasks of matrix inversion or eigenvalue decomposition which scale as $O(n^3)$. Moreover, those kernel algorithms which use tools such as semi-definite programming have even higher-order polynomial complexities [10]. Nyström based methods, have been shown to be efficient techniques for the eigenvalue decomposition of large kernel matrices [11], [8], [12]. For example, the clustered Nyström method [8] employs an efficient approach to computing matrix approximations with a high degree of accuracy. A common approach when using feature extraction and dimensionality reduction algorithms involves passing the Gram matrix as input into the classifier [11], [8].

The work proposed in this paper describes and experimentally evaluates an approach of using the Singular Value Decomposition and the Nyström matrix approximation methods [8] as approaches for extracting features which will be used for classification tasks, without constructing the Gram matrix. This means using the matrix containing the leading eigenvectors, and avoiding the computationally expensive task of constructing a kernel matrix by computing an inner product of feature vectors. The proposed approach is demonstrated using biomedical datasets, however, any type of dataset which has been prepared for classification tasks (i.e. inputs and labels) can be used. The approach described in this paper is different to the approach proposed by Zhang et al. [11], [8] who experimented with various approximation methods for classification tasks using Gram matrices.

The paper is structured as follows: Section II discusses related works; Section III describes related manifold learning and low-rank approximation methods; Section IV describes the problem definition and hypotheses; Section V provides the proposed method and architecture; Section VII discusses the experiments performed using the proposed architecture and a number of datasets which are typically adopted in biomedical research. Finally, Section X provides a conclusion and future work.

II. RELATED WORKS

Li et al. [12] argue that on very large datasets, the standard Singular Value Decomposition (SVD) algorithm takes $O(n^3)$ time, and it can become prohibitive in large-scale computations. Instead they proposed a Large-Scale Nyström Kernel matrix approximation using Randomized SVD, that initially samples a large column subset from the input matrix, but then only performs an approximate SVD on the inner submatrix using the recent randomized low-rank matrix approximation algorithms. Using the same arguments as Li et al. [12], Zhang et al. [11] proposed an Improved Nyström Low-Rank Approximation method. They compared the Improved Nyström Low-Rank Approximation method with state-of-the-art approaches that range from greedy schemes to probabilistic sampling, and found that their proposed Nyström achieved significant performance gains in a number of supervised/unsupervised learning tasks including kernel Principal Component Analysis and least squares Support Machines. In order to fit low-rank approximations into classification applications, they proposed the reconstruction of the eigen-system of a matrix approximated by its low-rank decomposition [11]. This common step in kernel methods is the reduction of the data to a kernel matrix, also known as a Gram matrix. The Gram matrix is then utilised for training and validating a machine learning classifier [9]. Similar to the work of Li et al. [12], Zhang et al. [11] have used the Gram matrices which resulted from the approximation methods, as input into the machine learning model for performing the classification task. Zhang et al. [11] have applied their Nyström method to 8 datasets and analysed approximation errors as an indication of performance, however, such analysis alone cannot be an indication of how the approximation method will perform when its outputs are fed into a classifier. The authors present limited results on the classification accuracy of the approximation methods under scrutiny. They only present the results of applying the methods coupled with the Support Vector Machine classifier on one dataset, the USPS digits (US Postal Service Dataset) dataset, where SVD outperformed their Nyström method on 6 out of 11 testing USPS tests datasets. However, one can conclude that the performance of SVD and

Nyström was approximately similar for that specific dataset. Their experiments also revealed that SVD was slower than their proposed Nyström method. The reason that SVD took longer than their Nyström method is due to the fact that Gram matrices were used in their experiments and to derive these matrices for SVD it is more computationally complex than with Nyström. Nevertheless, Nyström methods appear to be a good alternative to SVD, and one of the aims of this paper is to perform further experiments to appropriately compare and reach a steady conclusion on performance differences by means of accuracy and time. Li et al. [12] explain that computing the kernel matrix involves quadratic space complexity, together with the often-involved cubic time complexity, and this can be demanding in large-scale and big data applications. Li et al. [12] propose that a useful approach to reduce the computational burdens of computing the Gram matrix K, where given a set of m samples the kernel matrix K is of size $m \times m$, is to perform low-rank approximation [12], [11]. This involves, approximating K by $G \times G^T$, for some $G \in R^{m \times k}$. With $k << m$, the complexities associated in the handling of matrix G are much lower than those with matrix K.

This paper, presents experiments to explore the use of matrix G for classification tasks with more details on this presented in Section IV.

III. RELATED METHODS

Low-rank approximation is used in manifold learning and dimensionality reduction algorithms that rely on the eigenvectors of the kernel matrix [11]. The aim of low-rank matrix approximation is to obtain more compact representations of the data with limited loss of information, and using fewer dimensions than the original data [13]. Therefore, low-rank approximation methods construct an approximation of the original matrix which has a rank less than the rank of the original matrix. This section summarises the related methods for constructing low-rank approximation of matrices and describes the concept of Kernel Spectral Clustering, Singular Value Decomposition, and the Improved Nyström and Random Sampling Nyström approximation algorithms.

A. KERNEL SPECTRAL CLUSTERING

Spectral clustering algorithms exploit pairwise similarities of data instances. Liu et al [14] proposed a Spectral Ensemble Clustering (SEC) algorithm via weighted k-means clustering. They proposed Spectral Ensemble Clustering (SEC) to make use of the advantages of co-association matrix and have applied the proposed method to ensemble and multi-view clustering tasks. The authors state that the purpose of multi-view clustering is to separate instances into different groups based on multiple representations. Kernels have several meanings, and this paper follows the Mercer (positive definite) kernel. The methods studied in this paper require that the kernel function satisfies the requirement that the Gram matrix, X, be a positive definite for any set of inputs $x_{i=1}^N$. Let $x_i \in \Re^D$ be a vector of matrix X. Let $k(x, x') \geq 0$ be some measure of similarity between objects $x, x' \in X$, and k is a kernel function. Such a matrix is referred to as a Mercer kernel, or a positive definite kernel. Given matrix X, the eigenvector decomposition of matrix X can be computed using $X = U^T \Lambda U$, where Λ is a diagonal matrix of eigenvalues $\lambda_i > 0$. The entries in the kernel matrix can be computed by performing an inner product of feature vectors. A major problem for kernel-based predictions (such as Support Vector Machines (SVMs) and Gaussian processes) is that they are computationally expensive with regards to finding solution scalings such as $O(n^3)$, where n is the number of training examples. One approach to reduce computational complexity is to perform low-rank approximation in order to reduce the dimensionality of the matrices.

B. SINGULAR VALUE DECOMPOSITION

Let X be an arbitrary matrix of size $m \times n$, where m is the number of vector rows, and n is the number of vector columns. Let $X_{i,j}$, $i = 1 \ldots m$, be the i^{th} row vector of matrix X, and $j = 1 \ldots n$ the j^{th} column vector of matrix X. Let rank $rank(X) = r$, the Singular Value Decomposition of matrix X be denoted as $X = U_r \times \Sigma_r \times V_r^T$, where V_r^T is the transpose of matrix V_r, Σ_r is a diagonal matrix containing the singular values of matrix X sorted in decreasing order, and U_r and V_r have orthogonal columns that contain the left and right singular vectors of matrix X corresponding to its singular values. Let X_k be the best rank-k approximation to X which has been reconstructed via $X_k = U_k \times \Sigma_k \times V_k^T$, where V_k^T is the transpose of matrix V_k, Σ_k is a diagonal matrix containing the k leading singular values of matrix X, and the U_k and V_k have orthogonal columns that contain the leading k left and right singular vectors of matrix X corresponding to its singular values. The aim is to generate an approximation $\widetilde{X}$ of matrix X based on a sample $k \ll n$. It is important to identify a suitable number of k dimensions to retain and which are needed to create a good approximation of the original matrix with fewer dimensions and minimum error.

C. NYSTRÖM MATRIX APPROXIMATION

The Nyström method is a technique for finding numerical approximation to eigenvalue decomposition. There exist several variants for Nyström approximation based spectral clustering. The Nyström method is used to generate low-rank matrix approximations and has been applied to several large-scale applications which require solutions to dealing with the high computational complexity of large datasets [1]. The most important step of the Nyström method is sampling, by choosing different sampled landmark points λ to obtain different approximations of the original matrix, and thus to approximately compute the kernel eigenfunctions. Uniform sampling without replacement is a popular approach for this purpose, where every point has the same probability of being included in the sample. The aim is to generate an approximation $\hat{X}$ of matrix X based on a sample of its columns n obtained from matrix X such that, $\lambda \ll n$.

The Nyström method approximates the full kernel matrix X by first sampling n columns, denoted by $\hat{x_1} \ldots, \hat{x}_n$. The Nyström method has shown to be a good solution toward finding numerical approximations to eigenfunction problems. Zhang et al. [11] proposed an Improved version of the Nyström approximation method which is based on a clustered data model that uses an alterned k-means classifier – which they named Effective k-Means. They proposed the use of clustering algorithms to naturally obtain the data clusters by assigning each data sample to its closest landmark point.

D. RANDOM SAMPLING USING LANDMARK POINTS

Williams and Seeger [15] proposed a Random Sampling technique for the Nyström method to speed up computing kernel eigenfunctions. Random Sampling works by choosing a subset of samples, called landmark points, to construct a low-rank matrix approximation by computing the kernel eigenfunctions. Thus, it approximates matrix $\hat{K}$ by randomly choosing m rows/columns of K (without replacement). The random sampling technique proposed by Williams and Seeger [15] for approximating matrix X, gives rise to $O(m^2n)$ computational complexity. The quality of the Nyström approximation depends significantly on the subset of columns used, which are usually selected using random sampling. Choromanska et al. [16] proposed a fast spectral clustering algorithm with computational complexity linear in the number of data points that is directly applicable to large-scale datasets. However, a significant obstacle to scaling up spectral clustering to large datasets is that it requires building a similarity matrix between pairs of data points which becomes computationally expensive for high dimensional datasets, that is datasets which have a large number of features.

IV. PROBLEM DEFINITION

As a preliminary, let matrix X be a, $m \times n$, case-by-feature matrix. Since the paper is concerned with biomedical and health data classification, in the datasets herein $m \times n$ matrices are patient-by-feature matrices. Therefore, each of the datasets utilised in the experiments comprise of 1) a patient-by-feature matrix, $X_{m \times n}$, and 2) a vector $Y_{m \times 1}$ where each element y_i holds the label for each row x_i of matrix X. Given the significant computational complexity associated with manipulating Gram matrices, the main hypothesis of the paper is that for machine learning classification tasks, there is no improvement in time and accuracy when using the Gram matrix K, which represents the case-by-case matrix, and in the context of this paper the patient-by-patient matrix, as input into the classifier. Instead, the reduced patient-by-dimension matrix G_k can be directly input into a machine learning classifier without having to compute the Gram matrix. The subsections that follow explain how the SVD and Nyström methods will be adopted for exploring the main hypothesis, and the remaining paper focuses on exploring this hypothesis on biomedical datasets.

A. EIGENVECTORS FROM NYSTRÖM METHODS FOR CLASSIFICATION TASKS

The Nyström method can be utilized for obtaining orthogonal eigenvectors [11]. Zhang et al. [11] proposed an Improved Nyström method based on clustering algorithms, and for classification tasks their evaluations involved using the approximation of the original matrix. Given $G \in \Re^{n \times m}$ is a lower triangular matrix, and $m \ll n$, the top m eigenvectors U of the original patient-by-feature matrix X can be obtained as $U \approx GV\Lambda^{1/2}$ in $O(m^2n)$ time, where $V, \Lambda \in R^{m \times m}$ are from the eigenvalue decomposition of the $m \times m$ matrix $S = G^TG = V\Lambda V^T$ [11].

Hypothesis 1: For classification tasks the lower triangular matrix G can be used as input into a classifier without the need to compute the matrix K via $G \times G^T$. Matrix K is of size $m \times m$, and it is known as the Gram matrix. The eigenvectors found in matrix G can be used as input into a machine learning classifier without compromising accuracy, compared to using the K matrix.

Once the Nyström algorithm is applied to the patient-by-feature matrix X, and the cluster centres c_j are derived, then each cluster centre becomes a landmark point λ_i. The aim is to have the smallest number of landmark points needed to represent every sample accurately (this is the task of dimensionality reduction). This is a challenging task to do, particularly when the dataset contains a lot of noise, and also a high degree of uncertainty. The performance of the Nyström matrix approximation algorithms depends on the number of landmark points chosen, and the optimal number of landmark points is decided by observing the classifiers accuracy and speed. The aim is to have high performance (measured by high classification accuracy achieved by the classifier in least time) using a minimum number of landmark points, since increasing the landmark points increases the dimensionality of the matrix which is input into the classifier and therefore this impacts on processing time.

B. EIGENVECTORS FROM SINGULAR VALUE DECOMPOSITION FOR CLASSIFICATION TASKS

Given X, an $m \times n$ patient-by-feature matrix, and $s = min(m, n)$. If X has a rank r, then there is an $m \times m$ unitary matrix G, an $n \times n$ unitary matrix V, and an $m \times n$ diagonal matrix $\Sigma = diag(\sigma_1, \ldots, \sigma_s)$ such that $X = G\Sigma V^T$ where $\sigma_1 \geq \ldots \sigma_r > 0 = \sigma_{r+1} = \cdots = \sigma_s$. The scalars $\sigma_1, \ldots, \sigma_s$ are called singular values and are the square roots of the non-zero eigenvalues of X^TX, ordered by size. The Singular Value Decomposition can be used to reveal the rank (r_X) of matrix X. (r_X) of matrix X is the number of nonzero diagonal elements of Σ. A rank-k approximation to matrix X, denoted as (X_r), can be defined where $k \leq r_X$, by setting all but the k-largest singular values of X equal to zero. The patient-by-patient matrix K can be reconstructed via $K = (G \times \Sigma) \times (G \times \Sigma)^T$, where K is $m \times m$, and it is considered as the Gram matrix.

Hypothesis 2: For classification tasks the lower triangular matrix G_k can be used as input into a classifier without the

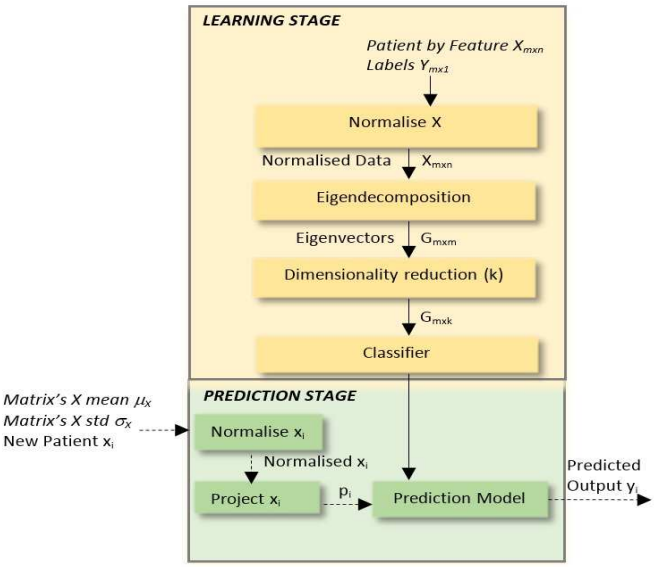

FIGURE 1. Illustration of the proposed architecture for using approximation methods to extract data for training a classifier. $G_{m\times k}$ is the patient-by-dimension matrix.

need to compute matrix K via $(G_k \times \Sigma_k) \times (G_k \times \Sigma_k)^T$.

When applying Singular Value Decomposition on a matrix, its performance depends on the number of k dimensions chosen. Choosing a larger number of dimensions than needed can result in including noise in the dataset, and choosing too few dimensions may remove important information.

V. PROPOSED METHOD

This section describes the proposed architecture illustrated in Fig. 1 and explains how to utilise a prediction model which has Eigendecomposition and dimensionality reduction components to predict outcomes of new records, which in this paper are patient records. In particular, a patient record can contain n number of features, where features can be clinical, gene expression, flow cytometry immunophenotyping, biomedical and other data. This paper focuses on biomedical and healthcare data obtained from body sensors.

Input: Let X be a pre-processed $m \times n$ patient-by-feature matrix. Pre-processing can include imputation of missing values and preparation of dataset.

Normalise: In order to improve prediction results, it is best to normalise the training dataset, X, with the preferred normalisation function. In this paper the zscore normalisation function is adopted, however zscore can be replaced by any other alternative normalisation function. For a sample data with mean $\bar{X}$ and standard deviation S, the zscore of a data point is $z = (x - \bar{X})/S$. The zscore transformation measures the distance of a data point from the mean in terms of the standard deviation. The normalised (or standardized) data set has a mean value of 0 and standard deviation 1, and retains the shape properties of the original data set (same skewness and kurtosis).

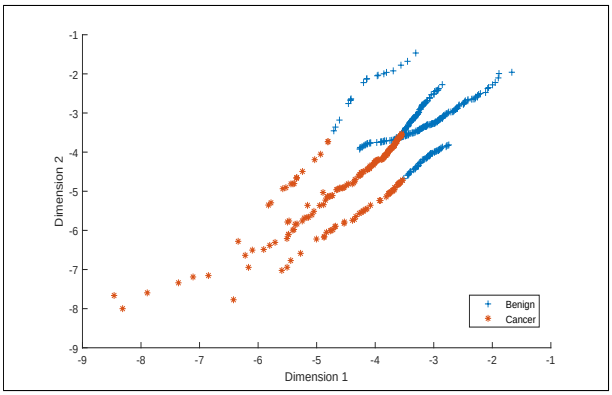

FIGURE 2. Example of SVD-G derived from applying SVD to the breast cancer dataset using $k = 2$ dimensions. SVD-G plotted with Benign and Cancer patient data shown in blue and red respectively. A logarithmic scale was used for better visualisation of the data.

Eigendecomposition and dimensionality reduction: are applied to matrix $X_{m\times n}$ using a pre-specified rank-k value if SVD is applied, or using a k value which is basically the number of landmark points λ if Nyström is applied. After applying SVD and dimensionality reduction, the result is a patient-by-dimension matrix $G_{m\times k}$, a Singular Values matrix, $\Sigma_{k\times k}$, and a feature-by-dimension matrix, $V_{n\times k}$. If Nyström is applied then a patient by dimension matrix $G_{m\times k}$ and a matrix C containing the k-means centroids used by the method are two of the matrices returned. In SVD and Nyström the values of k and λ respectively indicate the number of dimensions of matrix G. Figures 2 and 3 visualise the data from matrices G derived from applying SVD and the Effective k-means Nyström methods to the Breast Cancer dataset. Using SVD and Nyström, the data was reduced to 2 dimensions for visualisation purposes.

Classifier: A machine learning classifier is then trained using the truncated matrix $G_{m\times k}$. The output is a trained classification model (i.e. learned model) which can be used to predict the outcome of one or more new patient records, x_i. In this paper the k-Nearest Neighbor classifier was adopted, but any other classifier can be used.

Prediction Model: The prediction model takes a new record, p_i and predicts its class (e.g. cancer, benign disease, etc.).

VI. EXPERIMENTAL SETUP

The architecture proposed in Section V is adopted to evaluate the hypotheses provided in Section IV using biomedical datasets. Predictive performance is evaluated in terms of accuracy and time with and without using the Gram matrices derived from applying Singular Value Decomposition, and two versions of the Nyström approximation algorithm – Effective k-means Sampling, and Randomised Sampling Nyström. Experiments were performed using an Intel(R) Core (TM) i7CPU 3.3GHz, and 32 GB RAM.

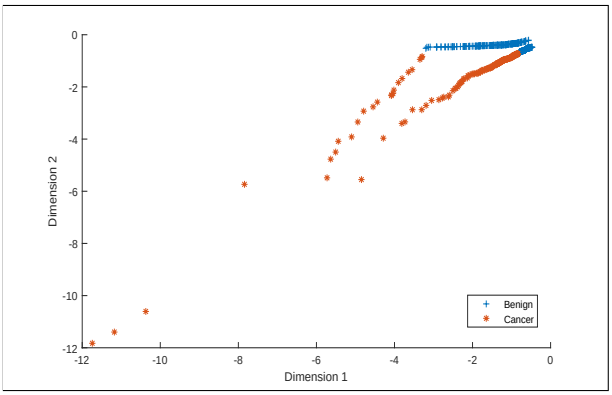

FIGURE 3. Example of Nyström-G derived from applying Nyström to the breast cancer dataset using $\lambda = 2$ landmark points. Nyström-G plotted with Benign and Cancer patient data shown in blue and red respectively. A logarithmic scale was used for better visualisation of the data.

Dataset characteristics			
	Cases	# Features	# Classes
Ovarian	216	100	2
Leukemia	72	7219	2
PANCAN	801	20531	5
BreastDiag	569	30	2
Hepatitis	155	19	2
Dermatology	366	34	6
Heart-c	303	13	2
Heart stat	270	13	2
Diabetes	768	8	2
Hypothyroid	3163	25	2
Hyperthyroid	3771	29	4
Thyroid(allbp)	3772	30	3
Appendicitis	106	7	2
GAMETES-Epistasis	1600	1000	2

TABLE 1. Benchmark datasets used in the experiments.

A. BIOMEDICAL DATASETS

Datasets were downloaded from online repositories containing high-dimensional biomedical datasets. Some of the datasets used in the experiments are benchmark biomedical datasets commonly used for evaluating pattern recognition algorithms, whilst other datasets have been used in biomedical papers. The datasets and their characteristics are shown in Table 1. Some of the datasets contained NaN values, and these were replaced with a weighted mean of the k nearest-neighbour columns as part of the normalisation process. The nearest-neighbour column is the closest column in Euclidean distance. If the corresponding value from the nearest-neighbour column is also NaN, the next nearest column is used.

B. MULTI-MODAL MOBILE HEALTH (MHEALTH) DATASET

The mHealth[1] (Mobile Health) dataset is a benchmark dataset for human behaviour analysis based on multi-modal body sensing. The mHealth dataset comprises body motion and vital signs recordings for ten volunteers of diverse profile while performing 12 physical activities: Standing still (1 min), Sitting and relaxing (1 min), Lying down (1 min), Walking (1 min), Climbing stairs (1 min), Waist bends forward (20x), Frontal elevation of arms (20x), Knees bending (crouching) (20x), Cycling (1 min), Jogging (1 min), Running (1 min), and Jump front back (20x). Sensors on each subject's chest, right wrist and left ankle were used to measure the motion experienced by diverse body parts, namely, acceleration, rate of turn and magnetic field orientation. All sensing modalities are recorded at a sampling rate of 50 Hz, which is considered sufficient for capturing human activity [17]. This dataset has been found to generalize to common activities of the daily living, due to the diversity of body parts involved in each activity (e.g., frontal elevation of arms vs. knees bending), the intensity of the actions (e.g., cycling vs. sitting and relaxing) and their execution speed or dynamicity (e.g., running vs. standing still). Data from the subjects carrying out the activities were collected in an out-of-lab environment with no constraints on the way these must be executed, with the exception that the subject should try their best when executing them [17], [18].

[1] mHealth dataset https://archive.ics.uci.edu/ml/datasets/MHEALTH+Dataset

C. EXPERIMENT METHODOLOGY

This section explains the experiment methodology for comparing classification performance with and without using the Gram matrix derived using 1) the Singular Value Decomposition, 2) the Improved Nyström method which uses a k-Means sampling procedure [11], and 3) the Random Sampling Nyström method [12] which uses Random Permutation sampling procedure. During the experiments, the output of the Singular Value Decomposition, and the two Nyström methods are separately input into a k-Nearest Neighbour classifier and performance is compared using the datasets specified in Section VI-A. Because the aim is to evaluate the data matrices which are input into the classifier rather than to identify the best classifier, only one machine learning classifier was adopted. Nevertheless, the proposed approach can be adopted using any machine learning classifier. A simple classifier, k-Nearest Neighbour set with 10 nearest neighbours, was adopted, and each experiment was run 30 times using 10-fold cross validation. Average 10-fold cross-validation over 30 runs is reported. The processes of applying the SVD and Nyström methods are provided in Algorithms (1) - (3). It is considered that the best approximation methods are those which allowed the k-NN classifier to achieve the highest classification accuracy, using the smallest rank (i.e. least number of landmark points λ in Nyström methods, and least number of k dimensions in Singular Value Decomposition). Increasing the rank of the matrices can often improve performance, however, it will also increase the time needed to compute the Gram matrices, and it will also increase the time the machine learning algorithm takes to learn the data. Thus for testing the hypotheses, matrix G will be constructed using various k matrix-ranks ranging from 5 to 50. Matrix G will be of size $m \times k$. Matrix G is input into the classifier along with the target vector, $Y_{m\times 1}$, which contains the target

value (i.e. ground truth) for each record. In the remainder of the paper, SVD stands for Singular Value Decomposition, EKM for Improved Nyström method which uses the Effective k-Means Sampling procedure, and RS for the Random Sampling Nyström method. In the experimental results the following hold:

- Let method SVD-G use the truncated patient-by-dimension matrix G_k, derived from Singular Value decomposition. Matrix G_k holds the truncated left eigenvectors. This method is provided in Algorithm (1).
- Let method SVD-GG use the patient-by-patient Gram matrix GG derived from computing $(G \times \Sigma) \times (G \times \Sigma)^T$. This method is provided in Algorithm (2).
- Let methods EKM-G and RS-G use the matrix G_k which is the patient-by-landmark matrix derived from the EKM and RS Nyström methods respectively. Matrix G holds the k left eigenvectors. The rank-k is equal to the selected number of landmark points, λ. This method is provided in Algorithm (3).
- Let methods RS-GG and EKM-GG use the Gram matrix GG derived from computing $(G_k \times G_k^T)$, where rank-k is the number of landmark points. This method is provided in Algorithm (4).

Algorithm 1: Calculate kNN (with 10 nearest neighbors) classification accuracy using SVD-G. Average accuracy over 30 iterations.

Result: acc
$[m, n] \leftarrow size(A)$
$z = 30$ //number of iterations
for $j = 1 : z$ **do**
  for $i = 1 : 50, k = i \times 5$ **do**
    $[G, S, V] \leftarrow svds(A, k)$
    $[acc_i] \leftarrow kNN(G, Y, 10)$
  end
  $[avacc_j] \leftarrow mean[acc]$
end

Algorithm 2: Calculate kNN (with 10 nearest neighbors) classification accuracy using the SVD-GG. Average accuracy over 30 iterations.

Result: acc
$[m, n] \leftarrow size(A)$
$z = 30$ //number of iterations
for $j = 1 : z$ **do**
  for $i = 1 : 50, k = i \times 5$ **do**
    $[G, S, V] \leftarrow svds(A, k)$
    $[GG] \leftarrow (G \times S) \times (G \times S)^T$
    $[acc_i] \leftarrow kNN(G, Y, 10)$
  end
  $[avacc_j] \leftarrow mean[acc]$
end

Algorithm 3: Calculate kNN (with 10 nearest neighbors) classification accuracy using either RS-G or EKM-G. Average accuracy over 30 iterations.

Result: acc
$[m, n] \leftarrow size(A)$
$z = 30$ //number of iterations
for *j=1:z* **do**
  for *i=1:50,* $\lambda = i \times 5$ **do**
    $[C, G] \leftarrow Nystrom(A, \lambda)$
    $[acc_i] \leftarrow kNN(G, Y, 10)$
  end
  $[avacc_j] \leftarrow mean[acc]$
end

Algorithm 4: Calculate kNN (with 10 nearest neighbors) classification accuracy using either RS-GG or EKM-GG. Average accuracy over 30 iterations.

Result: acc
$[m, n] \leftarrow size(A)$
$z = 30$ //number of iterations
for *j=1:z* **do**
  for *i=1:50,* $\lambda = i \times 5$ **do**
    $[C, G] \leftarrow Nystrom(A, \lambda)$
    $[GG] \leftarrow G \times G^T$
    $[acc_i] \leftarrow kNN(G, Y, 10)$
  end
  $[avacc_j] \leftarrow mean[acc]$
end

VII. RESULTS PART I: APPLYING AND COMPARING THE EIGENDECOMPOSITION METHODS ON VARIOUS BIOMEDICAL DATASETS

This section discusses the results when comparing the performance (accuracy and time) of the eigendecomposition methods applied to various biomedical datasets. Tables 2 and 3 show results after applying the eigendecomposition methods along with the mean, maximum and minimum accuracy achieved across the various ranks (average of 10-fold cross validation run 30 times) for each dataset (shown in Table 4). In SVD, the maximum number of k dimensions cannot exceed the number of n features, and therefore the value of k was reported accordingly as shown in Tables 2 and 3. For example in Table 5, dataset *Hepatitis* reports results for SVD up to $k = 15$ because the Hepatitis dataset has $n = 19$ features as shown in Table 1 which holds the characteristics of the datasets. Table 4 shows the mean average results of applying the eigendecomposition methods. The line chart in Fig. 4 shows the average performance of each method across all datasets at various dimensionalities. The Boxplot in Fig. 5 plots the average classification accuracy across all datasets at various dimensions. Each box holds 10 average values for dimensions $5, 10, \ldots 50$ for a method. The methods which use the Gram matrices (i.e. GG matrices) achieved on average

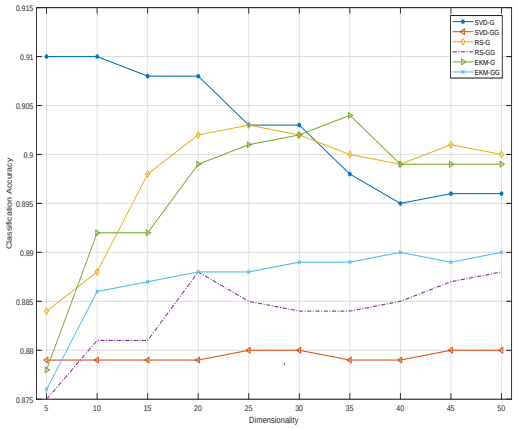

FIGURE 4. Line graph illustrating the average k-NN classification performance across the various dimensions. Approaches which did not use the Gram matrices achieved higher average classification accuracy than those which used Gram matrices. Average accuracy is derived from running each method 30 times. SVD-G needed fewer dimensions than other methods to achieve the highest classification accuracy across the datasets.

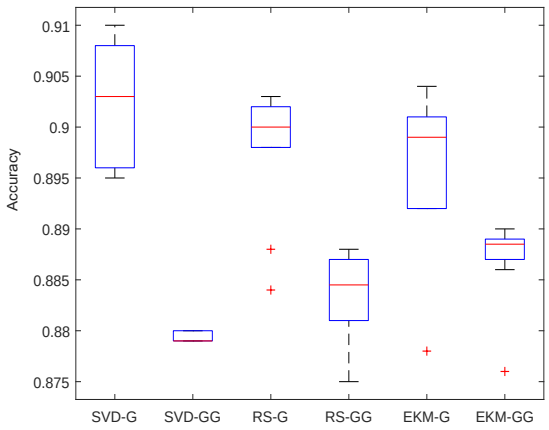

FIGURE 5. Average classification accuracy. Each box holds 10 mean accuracy values for dimensions $5, 10, \ldots, 50$ for a method. Average accuracy is derived from running each method 30 times. Approaches which did not use the Gram matrices achieved higher average classification accuracy as it can be seen by the higher position of the boxes.

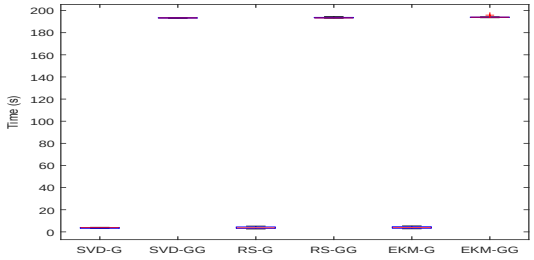

FIGURE 6. Average time shown across the various dimensions. Each box holds 10 average values for dimensions $5, 10, \ldots, 50$ for a method. Approaches which did not use the Gram matrices achieved lower decomposition and classification time as it can be seen by the lower position of the boxes.

lower classification accuracy. Observing the average performance of the methods across all datasets, when using SVD-G, the classifier returned the highest accuracy as expected, given that it is known to be a state-of-the-art eigendecomposition method. Overall, SVD-G achieved on average higher accuracy using fewer number of dimensions compared to the other methods achieving an average of 0.91 accuracy using 5-10 dimensions needing an average of 0.10 seconds; followed by the RS-G method which achieved 0.90 with 25 dimensions needing an average of 0.12 seconds. These details are shown in Table 4 along with the average accuracy and average time needed for the classifier when using each of the methods. Fig. 6 shows the average time taken for the classification task across various dimensions. The eigendecomposition and classification tasks were performed in much less time than when Gram matrices were not used.

The experimental results are consistent to support the hypotheses provided in Section IV. However, before reaching a final conclusion it is worth exploring whether there exist any significant differences in classification performance when using the SVD, EKM-Nyström and RS-Nyström methods.

A. ARE THERE ANY SIGNIFICANT DIFFERENCES IN CLASSIFICATION PERFORMANCE WHEN USING THE SVD, EKM-NYSTRÖM AND RS-NYSTRÖM METHODS?

Friedman's two-way Analysis of Variance (ANOVA) statistical test was adopted to determine whether there exist statistically significant differences between the mean values of the results obtained by the k-NN classifier when using the outputs of the various eigendecomposition methods across the various datasets (differences in terms of accuracy and time). Let $m \times n$ be a matrix A, where each cell a_{ij} holds the average performance value derived for Nyström landmark points/SVD using a particular method. Cells a_{ij} hold the average values (either accuracy or time) when using an eigendecomposition method (as illustrated in Table 4) at a particular dimensionality. Each column of the matrix A holds the results of the classifier when using one of 6 different approximation approaches. Friedman's chi-square statistic compares the mean values of the columns of matrix A. The test returned a statistically significant difference in performance depending on which output was input into the classifier, $\chi^2(5) = 41.39, p = 0.00$, and this suggests that the mean accuracy ranks of at least one approach is significantly different than the others.

A post-hoc multi-comparison test was run alongside the Friedman test to return the pairwise comparison results, and the results are shown in Table 5. The first two columns of Table 5 show the groups that are compared. Post hoc analysis was conducted with Bonferroni correction applied. Bonferroni adjustment was applied on the results because multiple comparisons are performed, and to reduce the likelihood of declaring a result as statistically significant when they should not be declared as such (a Type I error). The

Classification Results													
Ovarian							Leukimia						
	SVD		RS		EKM			SVD		RS		EKM	
k or λ	G	GG	G	GG	G	GG	k or λ	G	GG	G	GG	G	GG
5	0.947	0.944	0.939	0.913	0.921	0.928	5	0.831	0.824	0.853	0.806	0.841	0.840
10	0.940	0.944	0.939	0.931	0.930	0.934	10	0.814	0.834	0.799	0.818	0.831	0.828
15	0.935	0.942	0.937	0.931	0.944	0.935	15	0.810	0.833	0.820	0.795	0.767	0.848
20	0.922	0.941	0.948	0.934	0.949	0.938	20	0.825	0.829	0.840	0.834	0.805	0.840
25	0.913	0.946	0.943	0.937	0.947	0.938	25	0.814	0.837	0.822	0.798	0.798	0.842
30	0.918	0.944	0.944	0.940	0.966	0.940	30	0.823	0.835	0.854	0.795	0.803	0.827
35	0.895	0.946	0.952	0.936	0.954	0.937	35	0.800	0.831	0.824	0.801	0.829	0.828
40	0.876	0.943	0.939	0.936	0.948	0.936	40	0.794	0.829	0.818	0.804	0.800	0.851
45	0.884	0.942	0.954	0.936	0.957	0.937	45	0.806	0.833	0.776	0.809	0.806	0.833
50	0.877	0.944	0.936	0.936	0.951	0.938	50	0.800	0.832	0.814	0.847	0.814	0.840
Mean	0.911	**0.943**	**0.943**	0.933	**0.947**	0.936	Mean	0.812	**0.832**	**0.822**	0.811	0.809	**0.838**
Max	0.947	0.946	0.954	0.940	0.966	0.940	Max	0.831	0.837	0.854	0.847	0.841	0.851
Min	0.876	0.941	0.936	0.913	0.921	0.928	Min	0.794	0.824	0.776	0.795	0.767	0.827
PANCAN							BREAST						
	SVD		RS		EKM			SVD		RS		EKM	
k or λ	G	GG	G	GG	G	GG	k or λ	G	GG	G	GG	G	GG
5	0.994	0.983	0.939	0.960	0.982	0.986	5	0.951	0.948	0.966	0.952	0.949	0.948
10	0.997	0.988	0.996	0.966	0.995	0.996	10	0.937	0.950	0.962	0.955	0.959	0.953
15	0.994	0.988	0.993	0.988	0.995	0.996	15	0.921	0.948	0.961	0.954	0.963	0.952
20	0.995	0.988	0.993	0.992	0.995	0.996	20	0.914	0.948	0.961	0.954	0.962	0.954
25	0.996	0.988	0.994	0.992	0.998	0.996	25	0.899	0.949	0.964	0.953	0.963	0.953
30	0.995	0.987	0.994	0.991	0.994	0.996	30	0.887	0.949	0.957	0.954	0.960	0.955
35	0.993	0.988	0.993	0.992	0.996	0.996	35			0.966	0.953	0.958	0.955
40	0.996	0.987	0.995	0.994	0.992	0.997	40			0.956	0.953	0.958	0.954
45	0.997	0.988	0.994	0.995	0.994	0.996	45			0.955	0.955	0.955	0.955
50	0.997	0.987	0.993	0.995	0.997	0.996	50			0.957	0.954	0.955	0.955
Mean	**0.995**	0.987	**0.988**	0.987	**0.994**	0.995	Mean	0.918	**0.949**	**0.961**	0.954	**0.958**	0.953
Max	0.997	0.988	0.996	0.995	0.998	0.997	Max	0.951	0.950	0.966	0.955	0.963	0.955
Min	0.993	0.983	0.939	0.960	0.982	0.986	Min	0.887	0.948	0.955	0.952	0.949	0.948
Hepatitis							Derma						
	SVD		RS		EKM			SVD		RS		EKM	
k or λ	G	GG	G	GG	G	GG	k or λ	G	GG	G	GG	G	GG
5	0.887	0.877	0.872	0.863	0.851	0.853	5	0.946	0.769	0.759	0.725	0.695	0.686
10	0.896	0.878	0.896	0.877	0.886	0.867	10	0.972	0.766	0.808	0.756	0.845	0.766
15	0.889	0.878	0.883	0.846	0.875	0.856	15	0.969	0.764	0.889	0.762	0.877	0.774
20			0.890	0.866	0.888	0.869	20	0.966	0.764	0.929	0.783	0.907	0.791
25			0.884	0.853	0.891	0.861	25	0.953	0.767	0.920	0.785	0.923	0.797
30			0.865	0.859	0.885	0.863	30	0.926	0.768	0.931	0.782	0.949	0.795
35			0.855	0.863	0.883	0.868	35			0.941	0.790	0.948	0.800
40			0.883	0.864	0.880	0.864	40			0.932	0.787	0.957	0.801
45			0.893	0.863	0.883	0.862	45			0.953	0.788	0.935	0.795
50			0.879	0.865	0.883	0.861	50			0.946	0.782	0.943	0.797
Mean	**0.891**	0.878	**0.880**	0.862	**0.880**	0.862	Mean	**0.955**	0.766	**0.901**	0.774	**0.898**	0.780
Max	0.896	0.878	0.896	0.877	0.891	0.869	Max	0.972	0.769	0.953	0.790	0.957	0.801
Min	0.887	0.877	0.855	0.846	0.851	0.853	Min	0.926	0.764	0.759	0.725	0.695	0.686
Heart							Heartstat						
	SVD		RS		EKM			SVD		RS		EKM	
k or λ	G	GG	G	GG	G	GG	k or λ	G	GG	G	GG	G	GG
5	0.879	0.753	0.796	0.793	0.807	0.780	5	0.870	0.766	0.810	0.788	0.800	0.785
10	0.892	0.753	0.785	0.802	0.793	0.808	10	0.893	0.767	0.789	0.801	0.796	0.799
15			0.812	0.804	0.799	0.806	15			0.826	0.806	0.808	0.807
20			0.793	0.800	0.806	0.803	20			0.806	0.807	0.806	0.809
25			0.803	0.810	0.804	0.796	25			0.830	0.806	0.818	0.810
30			0.808	0.803	0.790	0.805	30			0.828	0.805	0.825	0.806
35			0.801	0.792	0.805	0.802	35			0.804	0.804	0.829	0.801
40			0.800	0.802	0.801	0.803	40			0.821	0.801	0.816	0.802
45			0.802	0.801	0.810	0.801	45			0.828	0.804	0.816	0.802
50			0.797	0.802	0.795	0.801	50			0.816	0.802	0.822	0.807
Mean	**0.885**	0.753	0.800	**0.801**	**0.801**	0.801	Mean	**0.881**	0.767	**0.816**	0.802	**0.814**	0.803
Max	0.892	0.753	0.812	0.810	0.810	0.808	Max	0.893	0.767	0.830	0.807	0.829	0.810
Min	0.879	0.753	0.785	0.792	0.790	0.780	Min	0.870	0.766	0.789	0.788	0.796	0.785

TABLE 2. Results of running 10-fold k-NN(k=10 nearest neighbours) 30 times on the patient-by-dimension matrix G, $m \times k$, and patient-by-patient matrix GG, $m \times m$ derived using SVD, Effective k-means and the Random-Sampling Nyström algorithms applied to each dataset. Results from classification using matrices G and GG.

fourth column shows the difference between the estimated group means. The third and fifth columns show the lower and upper limits for 95% confidence intervals for the true mean difference. The sixth column contains the p-value (adjusted after Bonferroni correction) for a hypothesis test that the corresponding mean difference is equal to zero.

The highlighted p-values are very small with $p < 0.05$, and these indicate that there are significant differences across all methods. Observing the pairs of particular interest, SVD-G and SVD-GG, RS-G and RS-GG, EKM-G and EKM-GG, there are significant difference between the mean values of the methods found in each pair (i.e. $p = 0.00$; $p = 0.00$, and $p = 0.004$ respectively for accuracy; and $p = 0.00$; $p = 0.00$, and $p = 0.00$ for time). The mean values of the methods SVD-G, RS-G and EKM-G where significantly higher than their corresponding methods SVD-GG, RS-GG and EKM-GG with regards to accuracy, and significantly lower with regards to time.

VIII. RESULTS PART II: CASE STUDY ON MULTI-MODAL MHEALTH DATA TO PREDICT HUMAN ACTIVITY FROM SMART PHONE DATA

This section describes the application of the proposed methods to extract features from a multi-modal mHealth dataset, described in Section VI-B. The mHealth dataset comprises 10 datasets, where each dataset holds the recordings of a single human participant. The rows of the datasets have been labelled as belonging to one of 13 classes where the first class 0 is the null class, and the remaining 12 classes correspond to human activities. The datasets are considered as 'Limited Training Datasets' [18] making this classification task a challenging one. The reason the datasets are considered to be "limited" is because the cases in the 'null' class range from 65.73%-78.19%, whereas the cases in the remaining

Classification Results													
Diabetes							Hypothyroid						
	SVD		RS		EKM			SVD		RS		EKM	
k or λ	G	GG	G	GG	G	GG	k or λ	G	GG	G	GG	G	GG
5	0.845	0.853	0.841	0.843	0.846	0.846	5	0.982	0.981	0.983	0.980	0.982	0.983
10			0.858	0.840	0.853	0.847	10	0.979	0.981	0.982	0.983	0.982	0.982
15			0.857	0.845	0.850	0.848	15	0.980	0.981	0.983	0.983	0.981	0.983
20			0.857	0.847	0.859	0.848	20	0.981	0.981	0.984	0.983	0.983	0.984
25			0.860	0.848	0.862	0.845	25	0.981	0.981	0.984	0.983	0.984	0.984
30			0.856	0.848	0.860	0.847	30			0.982	0.984	0.983	0.984
35			0.857	0.851	0.863	0.849	35			0.983	0.983	0.984	0.984
40			0.858	0.849	0.858	0.848	40			0.983	0.983	0.983	0.984
45			0.861	0.850	0.856	0.851	45			0.983	0.984	0.983	0.984
50			0.855	0.850	0.858	0.850	50			0.983	0.983	0.983	0.984
Mean	0.845	0.853	0.856	0.847	0.856	0.848	Mean	0.981	0.981	0.983	0.983	0.983	0.983
Max	0.845	0.853	0.861	0.851	0.863	0.851	Max	0.982	0.981	0.984	0.984	0.984	0.984
Min	0.845	0.853	0.841	0.840	0.846	0.845	Min	0.979	0.981	0.982	0.980	0.981	0.982
Thyroid							Hyperthyroid						
	SVD		RS		EKM			SVD		RS		EKM	
k or λ	G	GG	G	GG	G	GG	k or λ	G	GG	G	GG	G	GG
5	0.979	0.979	0.978	0.978	0.979	0.979	5	0.986	0.988	0.986	0.987	0.986	0.987
10	0.980	0.979	0.980	0.978	0.979	0.978	10			0.987	0.988	0.987	0.988
15	0.979	0.979	0.980	0.979	0.979	0.979	15			0.988	0.988	0.989	0.988
20	0.979	0.979	0.979	0.979	0.979	0.979	20			0.988	0.988	0.988	0.988
25	0.980	0.978	0.979	0.979	0.980	0.978	25			0.987	0.987	0.988	0.988
30	0.980	0.978	0.980	0.979	0.980	0.978	30			0.987	0.988	0.988	0.988
35			0.980	0.979	0.979	0.979	35			0.988	0.987	0.989	0.988
40			0.980	0.979	0.980	0.978	40			0.988	0.988	0.987	0.988
45			0.979	0.979	0.979	0.978	45			0.988	0.988	0.988	0.988
50			0.980	0.979	0.980	0.978	50			0.989	0.988	0.987	0.988
Mean	0.980	0.979	0.979	0.979	0.979	0.978	Mean	0.986	0.988	0.988	0.988	0.988	0.988
Max	0.980	0.979	0.980	0.979	0.980	0.979	Max	0.986	0.988	0.989	0.988	0.989	0.988
Min	0.979	0.978	0.978	0.978	0.979	0.978	Min	0.986	0.988	0.986	0.987	0.986	0.987
GAMETES							Appendicitis						
	SVD		RS		EKM			SVD		RS		EKM	
k or λ	G	GG	G	GG	G	GG	k or λ	G	GG	G	GG	G	GG
5	0.726	0.720	0.719	0.730	0.726	0.731	5	0.919	0.921	0.937	0.928	0.930	0.926
10	0.718	0.716	0.730	0.714	0.719	0.724	10	0.912	0.916	0.922	0.923	0.928	0.930
15	0.731	0.718	0.715	0.729	0.728	0.719	15	0.914	0.919	0.927	0.921	0.931	0.929
20	0.731	0.717	0.726	0.738	0.735	0.709	20	0.914	0.922	0.927	0.926	0.928	0.932
25	0.717	0.715	0.737	0.720	0.730	0.722	25	0.912	0.918	0.936	0.932	0.932	0.928
30	0.724	0.724	0.727	0.720	0.730	0.733	30	0.917	0.921	0.911	0.927	0.910	0.928
35	0.734	0.714	0.729	0.720	0.724	0.725	35	0.913	0.919	0.923	0.928	0.923	0.928
40	0.724	0.715	0.728	0.721	0.725	0.721	40	0.913	0.917	0.905	0.931	0.900	0.928
45	0.716	0.718	0.728	0.727	0.723	0.726	45	0.911	0.919	0.913	0.935	0.903	0.931
50	0.714	0.723	0.737	0.720	0.717	0.743	50	0.913	0.919	0.920	0.932	0.897	0.927
Mean	0.723	0.718	0.728	0.724	0.726	0.725	Mean	0.914	0.919	0.922	0.928	0.918	0.929
Max	0.734	0.724	0.737	0.738	0.735	0.743	Max	0.919	0.922	0.937	0.935	0.932	0.932
Min	0.714	0.714	0.715	0.714	0.717	0.709	Min	0.911	0.916	0.905	0.921	0.897	0.926

TABLE 3. Results of running 10-fold k-NN(k=10 nearest neighbours) 30 times on the patient-by-dimension matrix $m \times k$ which is matrix G, and patient-by-patient matrix $m \times m$ using RS, EKM and the SVD algorithms applied to each dataset. Classification using matrices G and GG.

Part I: Average Accuracy across all datasets						
	SVD		RS		EKM	
k or λ	G	GG	G	GG	G	GG
5	0.910	0.879	0.884	0.875	0.878	0.876
10	0.910	0.879	0.888	0.881	0.892	0.886
15	0.908	0.879	0.898	0.881	0.892	0.887
20	0.908	0.879	0.902	0.888	0.899	0.888
25	0.903	0.880	0.903	0.885	0.901	0.888
30	0.903	0.880	0.902	0.884	0.902	0.889
35	0.898	0.879	0.900	0.884	0.904	0.889
40	0.895	0.879	0.899	0.885	0.899	0.890
45	0.896	0.880	0.901	0.887	0.899	0.889
50	0.896	0.880	0.900	0.888	0.899	0.890
Av.	**0.903**	0.879	**0.898**	0.884	**0.897**	0.887
Max	0.910	0.880	0.903	0.888	0.904	0.890
Min	0.895	0.879	0.884	0.875	0.878	0.876
Part II: Average Time across all datasets						
	SVD		RS		EKM	
k or λ	G	GG	G	GG	G	GG
5	0.102	6.452	0.087	6.468	0.092	6.462
10	0.099	6.441	0.095	6.444	0.099	6.454
15	0.105	6.444	0.103	6.442	0.107	6.477
20	0.114	6.444	0.110	6.459	0.114	6.472
25	0.139	6.450	0.118	6.442	0.122	6.466
30	0.125	6.436	0.125	6.438	0.130	6.449
35	0.128	6.430	0.135	6.436	0.139	6.469
40	0.129	6.437	0.151	6.486	0.156	6.528
45	0.132	6.454	0.163	6.463	0.165	6.468
50	0.136	6.452	0.174	6.444	0.179	6.456
Av.	**0.121**	6.444	**0.126**	6.452	**0.130**	6.470
Max	0.139	6.454	0.174	6.486	0.179	6.528
Min	0.099	6.430	0.087	6.436	0.092	6.449

TABLE 4. Part I: Average Accuracy across all datasets. Each row holds 10 average values for k dimensions or λ landmark points (i.e. $5, 10, \ldots, 50$) for the SVD or Nyström methods. Average accuracy obtained with 10-fold cross validation executed 30 times across all datasets. Part II: Average Time across all datasets. The time reported is that of a single 10-fold cross validation, in order to demonstrate how long it takes to run the decomposition and classification process once. Results show that SVD-G was faster and more accurate than all other methods.

12 classes range from 0.67% -3.13%. Hence, the number of cases in all classes 1-13 are comparatively lower than the 0 class, and the number of training samples are limited.

Experiments were carried out to determine the best parameter settings for the SVD, EKM, and RS methods for the datasets. The results in Table 6, revealed that SVD-G achieved an average accuracy across the ten datasets of 0.94 (± 0.01) when using $k = 10$ dimensions, compared to the EKM-G which achieved an average accuracy of 0.88 (± 0.01), and RS-G achieved an accuracy of 0.90 (± 0.01) with $\lambda = 10$ landmark points. Table 7 shows the performance of each method for each class averaged across all 10 datasets when setting SVD to $k = 10$ dimensions, and Nyström to $\lambda = 10$ landmark points. It is also useful to observe the average results of each method for each class across the 10 datasets. SVD-G outperformed EKM-G and RS-G, where the methods achieved an average of 82.37%, 47.51%, and 61.0% respectively. Hence, a 34.86% improvement when using SVD-G instead of EKM-G, and a 21.37% improvement when using SVD-G instead of RS-G. The performance advantages of using SVD-G over the other approaches for the task of classifying data with limited training data are clear.

In order to provide a closer look at the results, the performance of the methods on Datasets 1 and 10, using confusion matrices are shown in Figs 7-12. The diagonal values show the number of correct classifications, and the off-diagonal

Multi-comparison test					
Comparison of means on Average Accuracy across all datasets					
		LL	MD.	UL	p
		$95\%CI$		$95\%CI$	
SVD-G	SVD-GG	0.016	0.023	0.031	**0.000**
RS-G	RS-GG	0.007	0.014	0.021	**0.000**
EKM-G	EKM-GG	0.002	0.009	0.017	**0.004**
SVD-G	RS-G	-0.002	0.005	0.012	0.593
RS-G	EKM-G	-0.006	0.001	0.008	1.000
SVD-G	EKM-G	-0.001	0.006	0.013	0.173
SVD-G	RS-GG	0.012	0.019	0.026	**0.000**
SVD-G	EKM-GG	0.008	0.016	0.023	**0.000**
SVD-GG	RS-G	-0.026	-0.018	-0.011	**0.000**
SVD-GG	RS-GG	-0.012	-0.004	0.003	1.000
SVD-GG	EKM-G	-0.024	-0.017	-0.010	**0.000**
SVD-GG	EKM-GG	-0.015	-0.008	-0.001	**0.026**
RS-G	EKM-GG	0.003	0.011	0.018	**0.001**
RS-GG	EKM-G	-0.020	-0.013	-0.005	**0.000**
RS-GG	EKM-GG	-0.011	-0.003	0.004	1.000
Comparison of means on Time taken to approximate and classify					
		LL	MD.	UL	p
		$95\%CI$		$95\%CI$	
SVD-G	RS-G	-1.042	-0.162	0.717	1.000
SVD-G	EKM-G	-1.167	-0.287	0.592	1.000
RS-G	EKM-G	-1.005	-0.125	0.755	1.000
SVD-G	SVD-GG	-190.571	-189.691	-188.812	**0.000**
RS-G	RS-GG	-190.658	-189.778	-188.898	**0.000**
EKM-G	EKM-GG	-191.075	-190.196	-189.316	**0.000**
SVD-G	RS-GG	-190.820	-189.940	-189.061	**0.000**
SVD-G	EKM-GG	-191.363	-190.483	-189.603	**0.000**
SVD-GG	RS-G	188.649	189.529	190.409	**0.000**
SVD-GG	RS-GG	-1.129	-0.249	0.631	1.000
SVD-GG	EKM-G	188.524	189.404	190.284	**0.000**
SVD-GG	EKM-GG	-1.671	-0.792	0.088	0.117
RS-G	EKM-GG	-191.200	-190.321	-189.441	**0.000**
RS-GG	EKM-G	188.773	189.653	190.533	**0.000**
RS-GG	EKM-GG	-1.422	-0.543	0.337	0.952

TABLE 5. Table shows the result of the statistical tests when comparing the accuracy and time taken to perform the decomposition and classification tasks using various methods and the Datasets in Table 1. Let M denote the method. The table shows that, in terms of accuracy, using the M-G and M-GG methods there are no significant differences in the performance of the k-NN classifier ($p < 0.05$), however, there are significant differences in time used to perform the decomposition and classification task (p=0.00).

Results using various methods on 10 mHealth Datasets												
SVD-G												
Dataset No.	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	Av.	Std
k												
2	0.88	0.86	0.85	0.85	0.85	0.84	0.83	0.87	0.86	0.84	0.85	0.02
4	0.89	0.87	0.86	0.87	0.87	0.87	0.86	0.88	0.88	0.86	0.87	0.01
6	0.90	0.90	0.90	0.90	0.90	0.91	0.91	0.90	0.92	0.90	0.90	0.01
8	0.93	0.93	0.92	0.92	0.93	0.93	0.93	0.93	0.93	0.93	0.93	0.01
10	0.95	0.94	0.94	0.93	0.94	0.95	0.94	0.94	0.94	0.94	0.94	0.01
Av.	0.91	0.90	0.89	0.89	0.90	0.90	0.89	0.91	0.91	0.89	0.90	0.01
Std.	0.02	0.03	0.03	0.03	0.03	0.04	0.04	0.03	0.03	0.04	0.03	0.00
EKM-G												
Dataset No.	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	Av.	Std
λ												
2	0.89	0.87	0.86	0.86	0.86	0.84	0.84	0.86	0.87	0.84	0.86	0.01
4	0.89	0.87	0.86	0.86	0.87	0.86	0.85	0.87	0.88	0.85	0.86	0.01
6	0.90	0.87	0.87	0.86	0.87	0.86	0.86	0.87	0.88	0.86	0.87	0.01
8	0.89	0.88	0.87	0.87	0.87	0.87	0.87	0.88	0.89	0.87	0.88	0.01
10	0.90	0.88	0.87	0.87	0.88	0.87	0.88	0.89	0.90	0.87	0.88	0.01
Av.	0.89	0.87	0.86	0.86	0.87	0.86	0.86	0.87	0.88	0.86	0.87	0.01
Std.	0.00	0.01	0.01	0.01	0.01	0.01	0.02	0.01	0.01	0.01	0.01	0.00
RS-G												
Dataset No./ λ	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	Av.	Std
λ												
2	0.89	0.87	0.86	0.85	0.86	0.84	0.84	0.86	0.87	0.84	0.86	0.02
4	0.90	0.87	0.87	0.86	0.87	0.87	0.87	0.87	0.88	0.85	0.87	0.01
6	0.91	0.89	0.88	0.87	0.88	0.89	0.88	0.89	0.90	0.89	0.89	0.01
8	0.91	0.89	0.88	0.89	0.89	0.90	0.90	0.90	0.90	0.87	0.89	0.01
10	0.91	0.90	0.91	0.89	0.90	0.91	0.89	0.91	0.92	0.91	0.90	0.01
Av.	0.90	0.88	0.88	0.87	0.88	0.88	0.88	0.89	0.89	0.87	0.88	0.01
Std.	0.01	0.01	0.02	0.02	0.01	0.03	0.02	0.02	0.02	0.03	0.02	0.00

TABLE 6. Table shows the results of comparing the various methods applied to ten mHealth datasets.

values show the number of incorrect classifications. The sum of correct and incorrect classifications are shown on the far right columns with % correct (in blue shades) and % incorrect (in red shades) respectively. The darker the shade the higher the value. As it can be observed from Figs 7-12, SVD-G achieved much higher accuracy than the alternative approaches with fewer misclassified cases. Finally, the experiments were run quickly on the mHealth datasets,

Av. Results across datasets (%)				
Class Description	Class No.	SVD-G	EKM-G	RS-G
Null class	0	91.15	91.18	90.95
L1: Standing still (1 min)	1	85.34	49.12	75.04
L2: Sitting and relaxing (1 min)	2	91.11	82.59	90.15
L3: Lying down (1 min)	3	94.57	92.44	94.84
L4: Walking (1 min)	4	81.16	35.37	50.07
L5: Climbing stairs (1 min)	5	77.22	22.54	38.64
L6: Waist bends forward (20x)	6	83.09	23.36	50.84
L7: Frontal elevation of arms (20x)	7	86.77	54.40	70.52
L8: Knees bending (crouching) (20x)	8	80.20	30.27	53.78
L9: Cycling (1 min)	9	67.29	31.02	46.60
L10: Jogging (1 min)	10	92.01	40.60	57.61
L11: Running (1 min)	11	85.79	58.64	63.89
L12: Jump front & back (20x)	12	55.12	6.05	10.02
Average		82.37	47.51	61.00

TABLE 7. Average classification accuracy for each class across all mHealth datasets. SVD to $k = 10$ dimensions, and Nyström methods were set to $\lambda = 10$ landmark points.

but the tasks of computing a Gram matrix and training a classifier using the Gram matrix took many hours. Therefore, only the results of the proposed approach, which is using the leading eigenvectors for training a classifier, are presented in this section. The performance comparison of using leading eigenvectors vs Gram matrix as input into the classifier has been explored in Section VII.

IX. DEEP SEQUENTIAL CLASSIFIER TO PREDICT HUMAN ACTIVITY USING THE MHEALTH DATA AFTER APPLYING THE PROPOSED FEATURE EXTRACTION EIGENDECOMPOSITION APPROACH

This section provides the results when using the SVD-G, RS-G, and EMK-G methods to extract features from ten large multi-modal multi-sensor mHealth datasets collected from a smart phone, to predict Human Activity using a Deep Sequential machine learning model. The structure of the model is shown in Figure 13. The SVD-G, RS-G and EKM-G approaches are compared and the results are shown in Table 8. Dimensionality for all methods was set to 10 dimensions, which is the same setting used in the previous experiments described in Section IX. Please note that due to the size of the datasets and the number of cases, conventional machine learning algorithms, besides kNN, were too slow to train, and for this reason they were not included in the results. Experiments were carried out using all 10 mHealth datasets described in Section VI-B, and 10-fold cross validation. Although the mHealth were large datasets and 10-fold would not normally be recommended for large datasets, 10-fold was suitable because the datasets are considered to be 'limited', as described in Section VIII.

As shown in Table 8 performing feature extraction using SVD-G resulted in higher classification accuracy compared to the other methods. SVD-G performed outperformed RG-G by 4.65% and outperformed EKM-G by 6.62%.

X. CONCLUSION

A common and important step in kernel methods is the reduction of the data to a kernel matrix, also known as a Gram matrix. The Gram matrix is then used for machine learning tasks such as classification and predictive modelling. A significant drawback of kernel methods is the computational

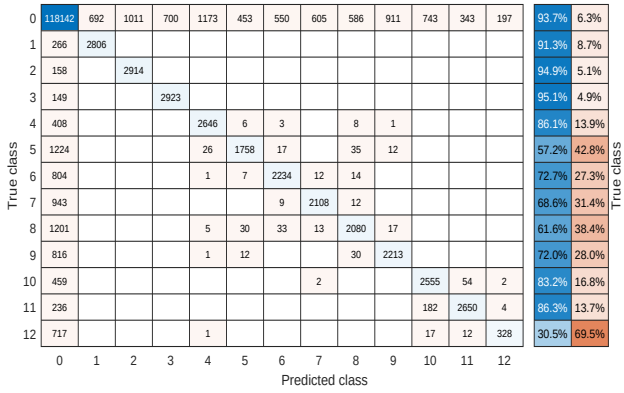

FIGURE 7. Dataset D1 SVD-G

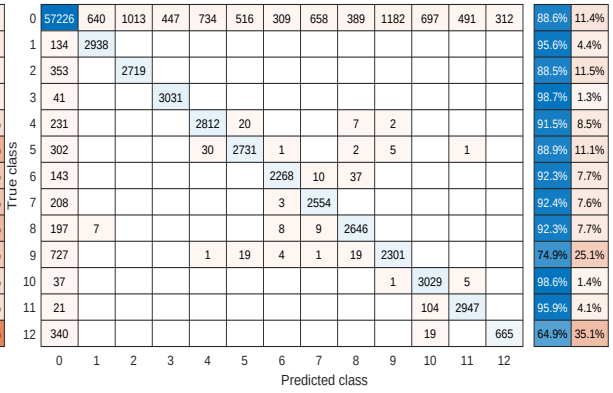

FIGURE 8. Dataset D10 SVD-G

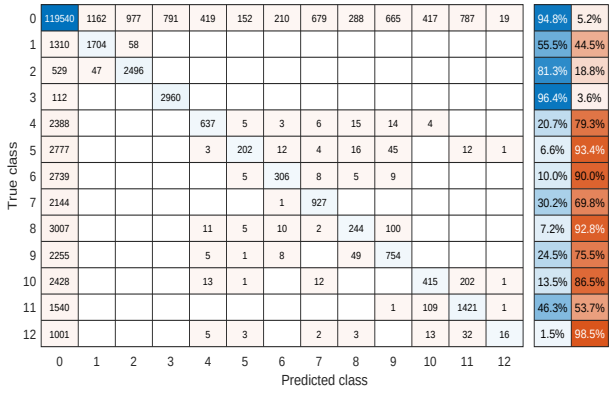

FIGURE 9. Dataset D1 EKM-G

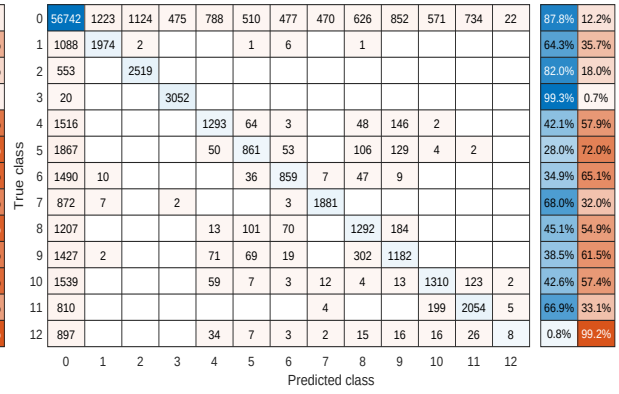

FIGURE 10. Dataset D10 EKM-G

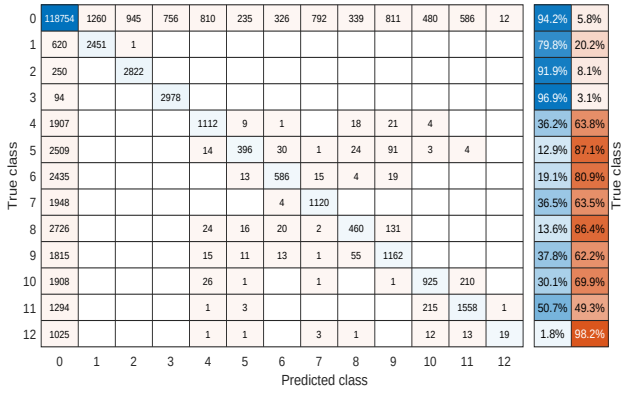

FIGURE 11. Dataset D1 RS-G

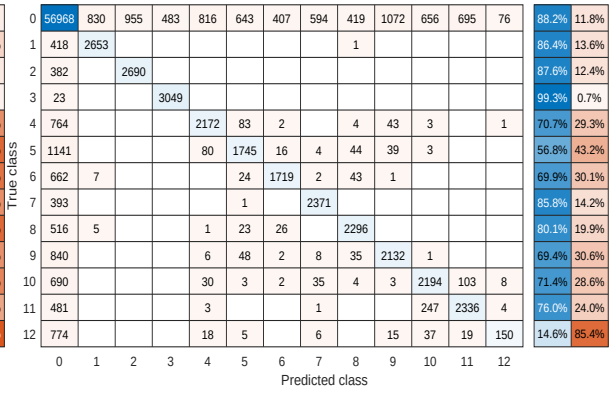

FIGURE 12. Dataset D10 RS-G

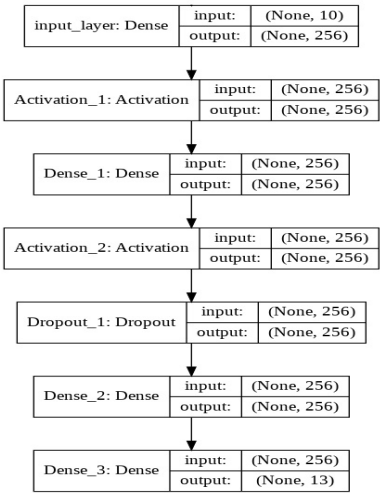

FIGURE 13. Sequential Model Structure

Deep Sequential Model for Behaviour classification						
	SVD-G		RS-G		EKM-G	
	Mean	Std	Mean	Std	Mean	Std
Dataset 1	82.29	9.61	75.16	10.14	76.82	5.87
Dataset 2	80.74	10.33	75.80	9.06	75.15	8.07
Dataset 3	79.27	7.63	76.68	6.62	75.11	5.29
Dataset 4	78.21	9.69	75.60	6.16	75.52	4.88
Dataset 5	81.86	14.02	77.88	12.62	72.97	9.48
Dataset 6	83.92	12.13	80.44	10.22	76.08	6.90
Dataset 7	82.55	7.97	77.41	4.45	75.41	4.27
Dataset 8	82.73	10.54	79.03	9.85	74.71	14.87
Dataset 9	85.98	10.22	82.16	8.53	79.31	6.14
Dataset 10	83.53	11.86	74.49	7.58	73.86	7.51
Average	82.11	10.40	77.47	8.52	75.49	7.33

TABLE 8. Deep Sequential Model. Table shows the results obtained when using the output of SVD-G, RS-G, and EKM-G as input into a Deep Learning approach, namely a Sequential model, for each of the mHealth datasets.

complexity associated with manipulating kernel matrices.

This paper proposes an efficient (in terms of time and accuracy) approach to apply eigendecomposition methods for classification tasks, without the need to construct Gram matrices. One limitation of Nyström methods compared to SVD, is that cluster centers are selected randomly in the Nyström method meaning that the approximation will need to be run several times to obtain a good overview about its effectiveness. On the other hand, SVD, does not suffer from this limitation, but it is computationally more complex. The approach is embedded into an architecture which can be adopted for biomedical and other predictive modelling tasks.

To evaluate the proposed architecture, experiments were conducted with 14 biomedical datasets to determine the effect on predictive accuracy and time when using the proposed approach. Experiments revealed significant differences in predictive performance in terms of accuracy ($p < 0.05$) and time ($p < 0.05$) when SVD and Nyström methods were used. A second experiment was carried out to compare the performance of the SVD and Nyström methods using multi-modal mobile health datasets for human behaviour analysis. The results revealed a 34.86% improvement when using the proposed SVD-G instead of EKM-G, and a 21.37% improvement when using SVD-G instead of RS-G.

The results provide evidence to support the main hypothesis of this paper, that the leading eigenvectors which represent the factor weights of each patient or person, need only be input into a classifier, and that there is no improvement in classification performance to construct and use a Gram matrix. Furthermore, the fact that when adopting the proposed approach, classification accuracy is higher on various datasets of different types (including multi-modal multi-sensor mHealth data) allows for the assumption that the improved accuracy is dependent on the solution of the approximation methods and thus the theoretical properties of the methods and not the datasets.

Future work includes applying the proposed approach to large image datasets using deep learning classifiers; and comparing the approach to more matrix approximation methods. The significance of the proposed classification and predictive modeling approach is that it can make feature extraction methods more accessible on large-scale data which is becoming common in many applications such as natural language processing, image processing, and other data analytics tasks where feature extraction is required.

REFERENCES

[1] S. Kumar, M. Mohri, and A. Talwalkar, "Sampling methods for the nyström method," J. Mach. Learn. Res., vol. 13, no. 1, pp. 981–1006, Apr. 2012.

[2] A. Hyvärinen and E. Oja, "Independent component analysis: Algorithms and applications," Neural Netw., vol. 13, no. 4-5, pp. 411–430, May 2000.

[3] M. Belkin and P. Niyogi, "Laplacian eigenmaps and spectral techniques for embedding and clustering," vol. 14, pp. 585–591, 2002.

[4] M. A. A. Cox and T. F. Cox, Multidimensional Scaling. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 315–347.

[5] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in Advances in Neural Information Processing Systems 14, T. G. Dietterich, S. Becker, and Z. Ghahramani, Eds., 2001, pp. 849–856.

[6] M. Fan, X. Zhang, H. Qiao, and B. Zhang, "Efficient isometric multi-manifold learning based on the self-organizing method," Inf. Sci., vol. 345, no. C, pp. 325–339, Jun. 2016.

[7] S. Mika, G. Ratsch, J. Weston, B. Scholkopf, and K. R. Mullers, "Fisher discriminant analysis with kernels," in Neural Networks for Signal Processing IX: Proceedings of the 1999 IEEE Signal Processing Society Workshop (Cat. No.98TH8468), Aug 1999, pp. 41–48.

[8] K. Zhang and J. T. Kwok, "Clustered nÿstrom method for large scale manifold learning and dimension reduction," IEEE Transactions on Neural Networks, vol. 21, no. 10, pp. 1576–1587, Oct 2010.

[9] F. R. Bach and M. I. Jordan, "Predictive low-rank decomposition for kernel methods," in Proceedings of the 22Nd International Conference on Machine Learning, ser. ICML '05. New York, NY, USA: ACM, 2005, pp. 33–40.

[10] G. R. G. Lanckriet, N. Cristianini, P. Bartlett, L. E. Ghaoui, and M. I. Jordan, "Learning the kernel matrix with semidefinite programming," J. Mach. Learn. Res., vol. 5, pp. 27–72, Dec. 2004.

[11] K. Zhang, I. W. Tsang, and J. T. Kwok, "Improved nyström low-rank approximation and error analysis," in Proceedings of the 25th international conference on Machine learning. ACM, 2008, pp. 1232–1239.

[12] M. Li, W. Bi, J. T. Kwok, and B.-L. Lu, "Large-scale nyström kernel matrix approximation using randomized svd," IEEE transactions on neural networks and learning systems, vol. 26 1, pp. 152–64, 2015.

[13] N. K. Kumar and J. Schneider, "Literature survey on low rank approximation of matrices," Linear and Multilinear Algebra, vol. 65, no. 11, pp. 2212–2244, 2017.

[14] H. Liu, J. Wu, T. Liu, D. Tao, and Y. Fu, "Spectral ensemble clustering via weighted k-means: Theoretical and practical evidence," IEEE Transactions on Knowledge and Data Engineering, vol. PP, no. 99, pp. 1–1, 2017.

[15] C. K. I. Williams and M. Seeger, "Using the nyström method to speed up kernel machines," in Advances in Neural Information Processing Systems 13, T. K. Leen, T. G. Dietterich, and V. Tresp, Eds. MIT Press, 2001, pp. 682–688.

[16] A. Choromanska, T. Jebara, H. Kim, M. Mohan, and C. Monteleoni, Fast Spectral Clustering via the Nyström Method. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 367–381.
[17] O. Banos, R. Garcia, J. A. Holgado-Terriza, M. Damas, H. Pomares, I. Rojas, A. Saez, and C. Villalonga, "mhealthdroid: A novel framework for agile development of mobile health applications," in Ambient Assisted Living and Daily Activities, L. Pecchia, L. L. Chen, C. Nugent, and J. Bravo, Eds. Cham: Springer International Publishing, 2014, pp. 91–98.
[18] L. T. Nguyen, M. Zeng, P. Tague, and J. Zhang, "Recognizing new activities with limited training data," in Proceedings of the 2015 ACM International Symposium on Wearable Computers, ser. ISWC '15. New York, NY, USA: ACM, 2015, pp. 67–74. [Online]. Available: http://doi.acm.org/10.1145/2802083.2808388