Eigen-Component Analysis: A Quantum Theory-Inspired Linear Model

Rongzhou Chen[1,*], Yaping Zhao[1,*], Hanghang Liu[2], Haohan Xu[2], Shaohua Ma[2,3], Edmund Y. Lam[1]
[1]*Department of Electrical and Electronic Engineering, The Univerisity of Hong Kong, Hong Kong SAR of China*
[2]*Tsinghua Shenzhen International Graduate School, Tsinghua University, Shenzhen, China*
[3]*Tsinghua-Berkeley Shenzhen Institute, Shenzhen, China*
[*]Co-first author
elam@eee.hku.hk

2025 IEEE International Symposium on Circuits and Systems (ISCAS) | 979-8-3503-5683-0/25/$31.00 ©2025 IEEE | DOI: 10.1109/ISCAS56072.2025.11044249

***Abstract*—In modern machine learning circuits and systems, the need for centralized, standardized data poses significant challenges, especially in privacy-sensitive, resource-constrained, real-time, or asynchronous environments. We introduce Eigen-Component Analysis (ECA), a quantum-inspired linear model that circumvents these requirements by leveraging intrinsic data structures. ECA uses an orthogonal transformation to extract meaningful features from decentralized, non-standardized data, ensuring high classification accuracy without extensive preprocessing. Unlike traditional linear models, ECA adapts to complex data distributions and shows superior performance in feature extraction and classification across various datasets. Experiments indicate that ECA achieves robust accuracy with fewer parameters, highlighting its potential where data centralization and standardization are impractical. Our work extends the feasibility of linear models in diverse scenarios, enhancing interpretability and efficiency in challenging data environments.**

***Index Terms*—Distributed Learning, Quantum-Inspired Algorithms, Linear Models, Feature Extraction, Classification**

I. Introduction

In modern circuits and systems with complex data structures, effective feature extraction and classification pose critical challenges in machine learning [1]. A fundamental problem is finding optimal methods to achieve high linear separability, essential for accurate classification across domains [2]. Conventional linear classifiers like Logistic Regression (LoR) and Linear Discriminant Analysis (LDA) perform well under conditions where data are centralized, standardized, and follow simple, linearly separable distributions [2]. However, real-world data often violate these assumptions due to privacy restrictions, decentralization, and non-uniform distributions, necessitating approaches to handle these complexities while preserving data integrity and interpretability [3]–[5].

Achieving reliable classification without centralized and standardized data is both crucial and challenging. In distributed or privacy-sensitive environments, such as healthcare and finance, aggregating data can be infeasible due to regulatory and logistical constraints [4], [5]. Additionally, non-standardized data with intricate structures often leads traditional linear models to fail, as they rely heavily on assumptions of uniformity and centralized processing [6]. Addressing these issues requires a method that handles distributed data while minimizing preprocessing like standardization, ensuring that classification models remain adaptable to dynamic, resource-constrained settings without sacrificing accuracy or interpretability.

Researchers have attempted to address these issues through various approaches. Some rely on dimensionality reduction techniques like Principal Component Analysis (PCA) or LDA, which effectively condense data but are limited by their dependence on centralized, standardized inputs [2]. Others, such as Quadratic Discriminant Analysis (QDA) and Support Vector Machine (SVM), offer better flexibility with non-linearity but tend to be computationally expensive and parameter-heavy, making them unsuitable for low-resource environments or real-time applications [7]. While these techniques perform well under controlled conditions, they often struggle with decentralized and heterogeneous datasets, limiting their scalability and practical applicability in complex real-world tasks [5].

To address these limitations, we propose Eigen-Component Analysis (ECA), a linear model inspired by quantum theory [8]. ECA introduces an orthogonal transformation that enables meaningful feature extraction from distributed, non-standardized data [9]–[11]. By leveraging the structural properties within the data, ECA circumvents the need for centralization or standardization, ensuring that classification occurs accurately across decentralized and diverse datasets. Our approach uses orthogonal transformations to map data into a feature space where class separability is maximized [9], [10]. Experiments demonstrate that ECA outperforms traditional methods in accuracy and requires fewer parameters, showcasing its potential in real-world applications.

Our main contributions are as follows:

- We propose ECA, a quantum-inspired model for linear separability, effectively addressing the limitations of decentralized, non-standardized, undersampled, and asynchronous data in real-world settings.
- We develop a robust orthogonal transformation that captures intrinsic data structures, enabling feature extraction without centralization or standardization.
- Experiments show ECA beats traditional classifiers, superior for complex data and resource-limited environments.

This work is supported in part by the Research Grants Council (GRF 17201822 and TRS T45-701/22-R) of Hong Kong SAR, China; and in part by Department of Science and Technology of Guangdong Province (2023B0909020003).

II. METHODS

We develop a classification framework inspired by quantum mechanics [12], [13], aiming to find a transformation matrix P that maps input data into a space where each class corresponds to a distinct group of features represented by a mapping matrix L. This section details the derivation of our method, starting from the problem formulation and leading to a practical algorithm implementable on classical computers.

A. Problem Formulation

Given a dataset $\{(\mathsf{x}^{(i)}, y^{(i)})\}_{i=1}^{n}$, where $\mathsf{x}^{(i)} \in \mathbb{R}^m$ is the input vector and $y^{(i)} \in \{1, 2, \ldots, l\}$ is its class label, our goal is to find an orthogonal transformation $P \in \mathbb{R}^{m \times m}$ such that the transformed features $\psi^{(i)} = P^\top \mathsf{x}^{(i)}$ can effectively discriminate between classes. The association between transformed features and classes is captured by a mapping matrix $L \in [0, 1]^{m \times l}$, where L_{jk} represents the degree of association between the j-th eigenfeature and class k.

B. Orthogonal Transformation

To ensure that the transformation preserves the structure of the data, we require P to be orthogonal, i.e., $P^\top P = I$, where I is the identity matrix. We achieve this by parametrizing P using the exponential of an antisymmetric matrix A [14]

$$P = e^A, \tag{1}$$

where $A \in \mathbb{R}^{m \times m}$ satisfies $A^\top = -A$. This parametrization ensures that P remains orthogonal during optimization.

ECA adopts a divide-and-conquer strategy by decomposing the multi-class classification problem into multiple binary classification problems at the level of eigenfeatures [15]. We consider each eigenfeature independently to determine its association with each class.

The mapping matrix L indicates whether an eigenfeature contributes to a particular class. We relax the binary constraint and allow L_{jk} to take continuous values in the interval $(0, 1)$, representing the probability of association [16].

C. Class Probability Estimation

Given an input vector $\mathsf{x}^{(i)}$, we obtain the transformed features $\psi_j^{(i)} = P^\top \mathsf{x}^{(i)}$ using the orthogonal transformation $P = e^A$. Each transformed feature $\psi_j^{(i)}$ corresponds to projections of the input data on each eigenfeature.

We compute the probability of $\mathsf{x}^{(i)}$ belonging to class k by summing the weighted contributions of all eigenfeatures

$$p_k^{(i)} = \sum_{j=1}^{m} |\psi_j^{(i)}|^2 L_{jk}, \tag{2}$$

where $|\psi_j^{(i)}|^2$ represents the magnitude squared of the j-th transformed feature, and L_{jk} is the degree of association between the j-th eigenfeature and class k. This mirrors the concept of probability amplitudes in quantum mechanics, where the magnitude squared of the amplitude reflects the likelihood of a particular state [17].

D. Objective Function

To derive the objective function, we aim to maximize the likelihood of the observed class labels $y^{(i)}$ for each input $\mathsf{x}^{(i)}$. We model the likelihood of observing the true class label $y^{(i)}$ for a given sample as the product of the estimated class probabilities

$$\mathcal{L}(\theta; \mathsf{x}, \mathsf{y}) = \prod_{i=1}^{n} \prod_{k=1}^{l} p_k^{(i) y_k^{(i)}}, \tag{3}$$

where $y_k^{(i)} = 1$ if the input belongs to class k, and 0 otherwise. Taking the logarithm of the likelihood, we obtain

$$\log \mathcal{L}(\theta; \mathsf{x}, \mathsf{y}) = \sum_{i=1}^{n} \sum_{k=1}^{l} y_k^{(i)} \log p_k^{(i)}. \tag{4}$$

Since we wish to minimize the negative log-likelihood, the final objective function becomes the cross-entropy loss [18]

$$\mathcal{J}(A, L) = -\sum_{i=1}^{n} \sum_{k=1}^{l} y_k^{(i)} \log p_k^{(i)}. \tag{5}$$

We optimize the parameters A and L to minimize this loss. The orthogonal transformation $P = e^A$ ensures that the transformation matrix remains valid, preserving the structure of the input data during optimization. We learn the association matrix L, bounded within $(0, 1)$, using a sigmoid function to ensure probabilistic interpretation.

E. Optimization Constraints

The optimization problem is constrained by

$$P = e^A, \quad A = A_{\text{raw}} - A_{\text{raw}}^\top + \text{diag}(D), \quad L_{jk} \in (0, 1). \tag{6}$$

We ensure P remains orthogonal during optimization by updating the trainable parameters A_{raw} and D, then computing A and P accordingly [19]. We keep the entries of L within $(0, 1)$ using a sigmoid function

$$L_{jk} = \sigma(\tilde{L}_{jk}) = \frac{1}{1 + e^{-\tilde{L}_{jk}}}, \tag{7}$$

where $\tilde{L}_{jk}$ are the unconstrained parameters. We adopt a Straight-Through Estimator (STE) [20] to approximate gradients to binarize or apply discrete constraints to L. This approach enables gradient flow through the thresholding operation, allowing L to remain interpretable while still being optimized using standard backpropagation techniques.

F. Implementation Details

We implement the ECA model using PyTorch [21], leveraging automatic differentiation and optimization tools. We define the antisymmetric matrix A as trainable parameters and compute the orthogonal transformation matrix $P = e^A$ using the matrix exponential. To ensure L remains in $(0, 1)$, we use a sigmoid activation function on its raw parameters $\tilde{L}$.

We train the model using the Adam optimizer with a learning rate of 0.01. Training is performed over 1000 epochs. The implementation of ECA is available at `https://github.com/lachlanchen/eca`.

G. Interpretability

The interpretability of the ECA model arises from the mapping matrix L, which shows the association between eigenfeatures and classes. Each entry L_{jk} quantifies the j-th eigenfeature's influence on class k, providing insights into data structure and feature relevance for classification.

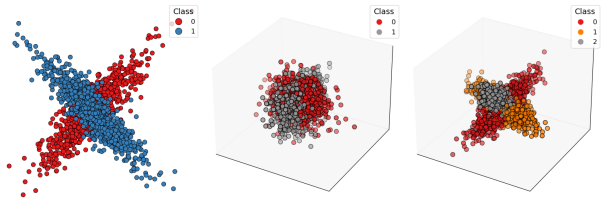

Fig. 1. Visualization of the synthetic 2D and 3D datasets.

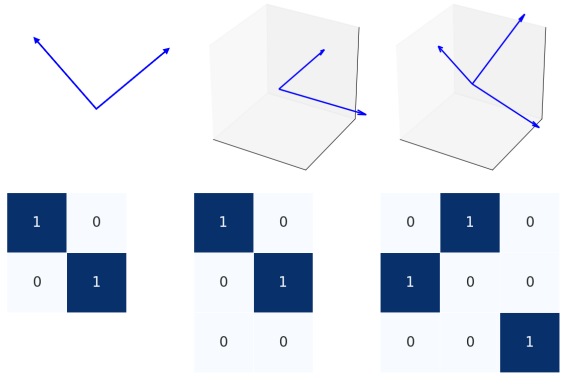

Fig. 2. Visualization of the learned transformation matrix e^A (top row) and mapping matrix L (bottom row) for the synthetic datasets using ECA.

III. Results

To demonstrate the effectiveness and interpretability of ECA, we compare it with traditional classifiers on datasets including synthetic 2D and 3D datasets, the Iris dataset [22], the Digits dataset [23], MNIST [24], and the Fashion MNIST [25] datasets. We compare our model with LoR, LDA, QDA, SVM, and Kernel Support Vector Machine (Kernel SVM) using the Radial Basis Function (RBF) kernel [2], [26], [27].

A. Synthetic Datasets

We evaluate our method on synthetic datasets designed to challenge linear classifiers due to overlapping features and covariance structures. The datasets include a 2D dataset with two classes, a 3D dataset with two classes, and a 3D dataset with three classes (see Figure 1).

In the 2D dataset, ECA achieves an accuracy of **86.00%**, outperforming others. The learned mapping matrix L associates each eigenfeature with a specific class, effectively capturing the class distinctions. The transformation matrix P finds the optimal rotation to enhance class separability (see Figure 2).

In the 3D dataset with two classes, ECA achieves an accuracy of **74.33%**. The model associates the first two eigenfeatures with the two classes, while the third eigenfeature is not associated with any class. This indicates that ECA identifies and discards the irrelevant feature, enhancing classification.

In the 3D dataset with three classes, ECA achieves an accuracy of **73.50%**. The mapping matrix L correctly associates each eigenfeature with one of the three classes, demonstrating ECA's capability in multiclass scenarios.

TABLE I
Comparison of Classifiers on Synthetic Datasets

Dataset	Model	Accuracy (%)	Parameters
2D	LoR	53.00	3
2D	LDA	52.33	3
2D	QDA	84.67	14
2D	SVM	56.33	3
2D	Kernel SVM	83.33	461
2D	ECA	**86.00**	9
3D (2 classes)	LoR	49.67	4
3D (2 classes)	LDA	49.67	4
3D (2 classes)	QDA	71.67	26
3D (2 classes)	SVM	48.67	4
3D (2 classes)	Kernel SVM	70.00	774
3D (2 classes)	ECA	**74.33**	15
3D (3 classes)	LoR	34.00	12
3D (3 classes)	LDA	33.83	12
3D (3 classes)	QDA	72.17	39
3D (3 classes)	SVM	47.33	12
3D (3 classes)	Kernel SVM	71.83	1,154
3D (3 classes)	ECA	**73.50**	18

TABLE II
Comparison of Classifiers on the Iris Dataset

Model	Accuracy (%)	Parameters
LoR	96.67	15
LDA	**100**	15
QDA	**100**	63
SVM	**100**	15
Kernel SVM	96.67	49
ECA	**100**	26

B. Iris Dataset

We apply ECA to the Iris dataset [22] to evaluate its performance on a well-known multiclass classification problem with four features and three classes. Additionally, we include a comparison with PCA [28] to provide context for the class separability achieved by ECA in contrast to traditional variance-based projections. Figure 3 displays the visualizations of both PCA and ECA projections for class comparisons on the Iris dataset, illustrating separations across selected eigenfeatures for Class 1 vs. 2, Class 2 vs. 3, and Class 1 vs. 3 based on the transformation matrix P.

The ECA model achieves an accuracy of **100%** on the test data, matching the performance of LDA, QDA, and SVM.

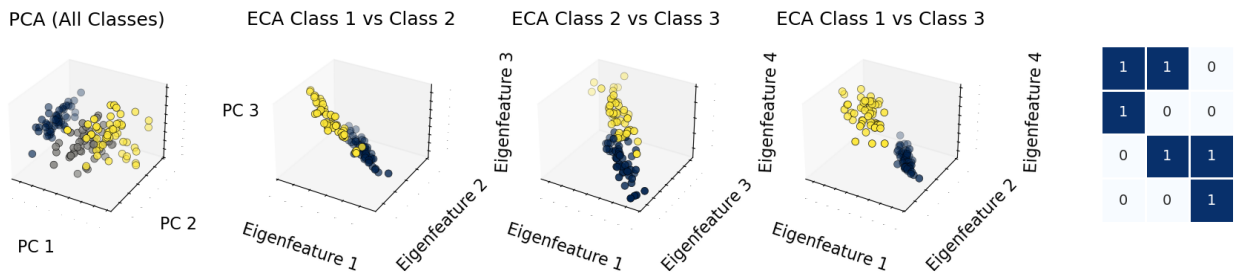

Fig. 3. Visualization of PCA and ECA projections for class comparisons on the Iris dataset. The figure presents a comparison between PCA (leftmost) and ECA projections across selected eigenfeatures, demonstrating class separations: Class 1 vs. 2 (second from left), Class 2 vs. 3 (third from left), and Class 1 vs. 3 (fourth from left), based on the transformation matrix P. PCA shows a variance-maximizing reduction, while ECA optimizes feature space for enhanced class-specific separability. The corresponding L matrix (rightmost) highlights the association between eigenfeatures and classes, providing interpretability.

C. Digits Dataset

We test ECA on the Digits dataset [23], which consists of 8×8 images of handwritten digits (total 64 features) across ten classes. The ECA model achieves an accuracy of **99.17%**, outperforming LoR, LDA, and SVM, and matching the performance of Kernel SVM.

TABLE III
COMPARISON OF CLASSIFIERS ON THE DIGITS DATASET

Model	Accuracy (%)	Parameters
LoR	86.67	650
LDA	95.28	650
QDA	73.33	41,610
SVM	88.06	2,925
Kernel SVM	**99.17**	663
ECA	**99.17**	2,784

D. Fashion MNIST Dataset

We apply ECA to the Fashion MNIST dataset [25], which consists of 28×28 grayscale images of clothing items across ten classes. We downsample the data by a factor of 100 due to convergence difficulties for traditional algorithms. The ECA model achieves an accuracy of **80.71%**, outperforming other linear models, while QDA suffered from colinearity issues in the high-dimensional, raw data.

TABLE IV
COMPARISON ON THE DOWNSAMPLED FASHION MNIST DATASET

Model	Accuracy (%)	Parameters
LoR	77.14	7,850
LDA	53.57	7,850
QDA	17.86	6,154,410
SVM	77.14	35,325
Kernel SVM	74.29	435
ECA	**80.71**	316,344

E. MNIST Dataset

We evaluate ECA on the MNIST dataset [24], which consists of 28×28 grayscale images of handwritten digits across ten classes. Due to convergence difficulties for traditional methods, we downsample the data by a factor of 100 (both train and test datasets). We also compare different algorithms with varying sample rates to observe their performance trends. The ECA model achieves an accuracy of **88.57%**, outperforming LoR and closely matching SVM.

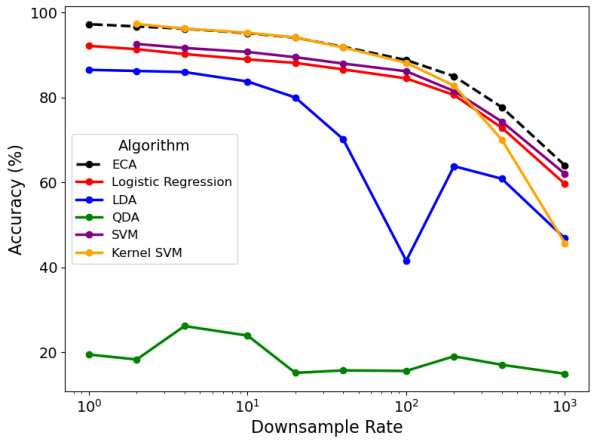

Fig. 4. Accuracy comparison on MNIST varying downsample fractions.

To further explore the robustness of ECA over different training sample sizes, we conduct experiments by varying the downsample fraction of the training data while keeping the test data ratio fixed (0.2 of the full dataset). We downsample the training data from 48,000 samples to 48 samples.

As shown in Figure 4, ECA consistently outperforms LoR and LDA, especially at lower downsample fractions. ECA maintains high accuracy even with significantly reduced training data, showcasing its robustness on sparse data.

IV. Conclusion

This paper introduces ECA, a quantum-inspired model designed to achieve linear separability without requiring data centralization or normalization. ECA leverages a robust orthogonal transformation to capture intrinsic data structures, enabling effective feature extraction and classification even in distributed and privacy-sensitive environments. Extensive experimentation across various datasets demonstrates that ECA consistently outperforms traditional linear classifiers, achieving high accuracy and efficiency, particularly in complex and resource-constrained settings.

References

[1] A. Labrinidis and H. V. Jagadish, "Challenges and opportunities with big data," *Proceedings of the VLDB Endowment*, vol. 5, no. 12, pp. 2032–2033, 2012.

[2] T. Hastie, "The elements of statistical learning: data mining, inference, and prediction," 2009.

[3] Q. Yang, L. Fan, and H. Yu, *Federated learning: Privacy and incentive*. Springer Nature, 2020, vol. 12500.

[4] G. A. Kaissis, M. R. Makowski, D. Rückert, and R. F. Braren, "Secure, privacy-preserving and federated machine learning in medical imaging," *Nature Machine Intelligence*, vol. 2, no. 6, pp. 305–311, 2020.

[5] Q. Li, Z. Wen, Z. Wu, S. Hu, N. Wang, Y. Li, X. Liu, and B. He, "A survey on federated learning systems: Vision, hype and reality for data privacy and protection," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 4, pp. 3347–3366, 2021.

[6] D. Singh and B. Singh, "Investigating the impact of data normalization on classification performance," *Applied Soft Computing*, vol. 97, p. 105524, 2020.

[7] D. Pavlov, J. Mao, and B. Dom, "Scaling-up support vector machines using boosting algorithm," in *Proceedings 15th International Conference on Pattern Recognition. ICPR-2000*, vol. 2. IEEE, 2000, pp. 219–222.

[8] E. Tang, "A quantum-inspired classical algorithm for recommendation systems," in *Proceedings of the 51st annual ACM SIGACT symposium on theory of computing*, 2019, pp. 217–228.

[9] M. Sanchez and J. Romagnoli, "Use of orthogonal transformations in data classification-reconciliation," *Computers & chemical engineering*, vol. 20, no. 5, pp. 483–493, 1996.

[10] P. P. Kanjilal and D. N. Banerjee, "On the application of orthogonal transformation for the design and analysis of feedforward networks," *IEEE Transactions on Neural Networks*, vol. 6, no. 5, pp. 1061–1070, 1995.

[11] J. Yen and L. Wang, "Simplifying fuzzy rule-based models using orthogonal transformation methods," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 29, no. 1, pp. 13–24, 1999.

[12] C. Ciliberto, M. Herbster, A. D. Ialongo, M. Pontil, A. Rocchetto, S. Severini, and L. Wossnig, "Quantum machine learning: a classical perspective," *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 474, no. 2209, p. 20170551, 2018.

[13] M. Schuld, I. Sinayskiy, and F. Petruccione, "An introduction to quantum machine learning," *Contemporary Physics*, vol. 56, no. 2, pp. 172–185, 2015.

[14] P.-A. Absil, R. Mahony, and R. Sepulchre, *Optimization algorithms on matrix manifolds*. Princeton University Press, 2008.

[15] T. G. Dietterich, "Ensemble methods in machine learning," in *International workshop on multiple classifier systems*. Springer, 2000, pp. 1–15.

[16] L. Liu, L. Wang, and X. Liu, "In defense of soft-assignment coding," in *2011 International Conference on Computer Vision*. IEEE, 2011, pp. 2486–2493.

[17] M. A. Nielsen and I. L. Chuang, *Quantum computation and quantum information*. Cambridge university press, 2010.

[18] J. A. Bilmes *et al.*, "A gentle tutorial of the em algorithm and its application to parameter estimation for gaussian mixture and hidden markov models," *International computer science institute*, vol. 4, no. 510, p. 126, 1998.

[19] J. Hu, X. Liu, Z.-W. Wen, and Y.-X. Yuan, "A brief introduction to manifold optimization," *Journal of the Operations Research Society of China*, vol. 8, pp. 199–248, 2020.

[20] Y. Bengio, N. Léonard, and A. Courville, "Estimating or propagating gradients through stochastic neurons for conditional computation," *arXiv preprint arXiv:1308.3432*, 2013.

[21] A. G. Baydin, B. A. Pearlmutter, A. A. Radul, and J. M. Siskind, "Automatic differentiation in machine learning: a survey," *Journal of machine learning research*, vol. 18, no. 153, pp. 1–43, 2018.

[22] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Annals of eugenics*, vol. 7, no. 2, pp. 179–188, 1936.

[23] E. Alpaydin and C. Kaynak, "Optical recognition of handwritten digits data set," *UCI Machine Learning Repository*, vol. 64, no. 5620, 1998.

[24] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[25] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms," *arXiv preprint arXiv:1708.07747*, 2017.

[26] C. Cortes, "Support-vector networks," *Machine Learning*, 1995.

[27] C. J. Burges, "A tutorial on support vector machines for pattern recognition," *Data mining and knowledge discovery*, vol. 2, no. 2, pp. 121–167, 1998.

[28] H. Abdi and L. J. Williams, "Principal component analysis," *Wiley interdisciplinary reviews: computational statistics*, vol. 2, no. 4, pp. 433–459, 2010.