



EEG SIGNAL CLASSIFICATION BASED ON ORTHOGONAL POLYNOMIALS, SPARSE FILTER, AND SUPPORT VECTOR MACHINE CLASSIFIER

Hayder S. Radeaf ^{1,2}, Mohammed Z. Al-Faiz ¹

¹ Department of Computer Engineering, College of Engineering, University of Baghdad, Iraq,

² College of Information Engineering, Al-Nahrain University, Baghdad, Iraq

haydersaadi@coeng.uobaghdad.edu.iq , mzalfaiz12@nahrainuniv.edu.iq

Corresponding Author: Mohammed Z. Al-Faiz

Received:22/11/2022, Accepted:15/08/2023

DOI:[10.31987/ijict.6.2.234](https://doi.org/10.31987/ijict.6.2.234)

Abstract- This work implements an Electroencephalogram (EEG) signal classifier. The implemented method uses Orthogonal Polynomials (OP) to convert the EEG signal samples to moments. A Sparse Filter (SF) reduces the number of converted moments to increase the classification accuracy. A Support Vector Machine (SVM) is used to classify the reduced moments between two classes. The proposed method's performance is tested and compared with two methods by using two datasets. The datasets are divided into 80% for training and 20% for testing, with 5-fold used for cross-validation. The results show that this method overcomes the accuracy of other methods. The proposed method's best accuracy is 95.6% and 99.5%, respectively. Finally, from the results, it is obvious that the number of moments selected by the SP should exceed 30% of the overall EEG samples for accuracy to be over 90%.

keywords: EEG, Orthogonal Polynomials, Sparse Filter, SVM, Tchebichef, Krawtchouk

I. INTRODUCTION

Brain-computer interface (BCI) technology enables people to control and interact with their surroundings [1]. BCI includes applications ranging from education and entertainment to environmental communication and control based on the evaluation of noninvasive EEG [2 - 4]. A prevalent paradigm for implementing a BCI system involves measuring EEG during distinct motor imagery tasks (e.g., left- and right-hand motor imagery). The EEG data collected during these motor imagery tasks are classified to control the BCI system.

Extracting the relevant features from the gathered EEG data is the key to producing control signals that will aid in categorizing different mental activities. Many techniques were used to extract features from different signals [5], [6].

Tchebichef polynomials (TP), Krawtchouk polynomials (KP), and Hahn polynomials are all examples of discrete orthogonal polynomials [7]. Although the TP shows notable energy compaction (EC) [8], the KP is superior to the TP in signal extraction of local features. On the basis that the Orthogonal Polynomial (OP) can be obtained by multiplying two orthogonal polynomials, numerous hybrid variants of KP and TP have been developed. Jassim et al. [9] proposed the Tchebichef-Krawtchouk polynomial (TKP), Mahmmud et al. [10] proposed the Krawtchouk-Tchebichef polynomial (KTP), and Abdulhussain et al. [11] recently offered the squared Krawtchouk-Tchebichef polynomial (SKTP). The SKTP is considered an advanced OPs combination strategy. It is implemented by multiplying two OPs derived from each hybrid OPs (TKP and KTP) [11].

Sparse filtering (SF) is a powerful technique in signal processing and machine learning that aims to extract informative features from high-dimensional data by encouraging sparsity in the representation [12]. Sparse filtering applies an unsupervised learning algorithm to learn a set of filters or basis functions that capture the most salient features of the input data. These filters are designed to be sparse, meaning only a few filters are activated for a given input. Sparse filtering has been successfully applied in various domains, including computer vision, natural language processing, and bioinformatics. It has shown promising results in tasks such as image denoising, feature extraction, and classification [13].

Support Vector Machines (SVM) are widely used machine learning algorithms that excel in classification and regression tasks. SVM aims to find an optimal hyperplane that maximally separates different classes in the feature space. SVMs can handle both linearly separable and nonlinearly separable data using kernel functions, enabling the input data to map into a higher-dimensional feature space. SVM has demonstrated excellent performance in various applications, including image recognition, text categorization, and bioinformatics [14].

In this paper, an EEG signal classifier is implemented. The classifier consists of converting EEG from a spatial domain to moments by using OPs then those moments are reduced by using a sparse filter. Finally, the reduced moments are fed to the SVM classifier to select the class to which this EEG signal belongs. The paper is organized as follows: The methodology section consists of theoretical concepts of OPs, SFs, and SVM classifiers. The Proposed EEG signal classifier section shows how the proposed method is implemented. The results and discussion section are shown next. Finally, a conclusion section summarizes the work.

II. THEORETICAL BACKGROUND

A. *Orthogonal Polynomials*

The signal (frame) is taken from the time (spatial) domain and transformed into the transform domain with the help of a transformation function, which is what a discrete transform is. Discrete transforms have many applications in signal processing, representation, and communication [15]. The powerful ability to analyze the elements of various signals [16] is greatly enhanced because it permits viewing signals in multiple domains. On top of that, orthogonal polynomials (OPs) have been intensively investigated and used in a wide range of applications [10]. OP generates Orthogonal Moments (OMs; see [11]). Because of their remarkable efficiency and effectiveness, QMs (scalar quantities) are used as a shape descriptor [17].

OPs include several types, such as the Krawtchouk polynomial (KP) [18] and the Tchebichef polynomial (TP) [8]. The OP kernels of the KP and TP are used to construct the discrete Krawtchouk transform (DKT) and the discrete Tchebichef transform (DTT). Krawtchouk-Tchebichef polynomials (KTP) [10] and Tchebichef-Krawtchouk polynomials (TKP) [9] are recently proposed hybrids of the original Krawtchouk and Tchebichef polynomials. Discrete Krawtchouk-Tchebichef transform (DKTT) and Tchebichef-Krawtchouk transform (DTKT) refer to transformations implemented by the KTP and TKP kernels, respectively. Signals are converted from the spatial to the moment domain using KTP and TKP kernels.

Comparative analyses of DTKT, DKTT, DKT, DTT, and DCT are presented in [9], and [10]. Compared to other real transforms, the KTP and TKP sets performed exceptionally well in energy compaction (EC) and localization properties [17]. This analysis demonstrated that DTKT and DKTT could use the localization property in the time and space domains

to extract features from the desired signal region. In addition, since these OPs redistribute signal energy into a limited number of polynomial coefficients, they exhibit substantial EC. Moments derived from an EEG signal using DTKT and DKTT are depicted in Fig. 1. The moment distribution shows they are more evenly distributed in the middle at DKTT and at the two ends at DTKT. Compared to the DTKT, the computational cost of extracting the test EEG signal's dominant moments (features) is drastically lower in the DKTT. Therefore, DKTT is employed here to extract features from the EEG signal.

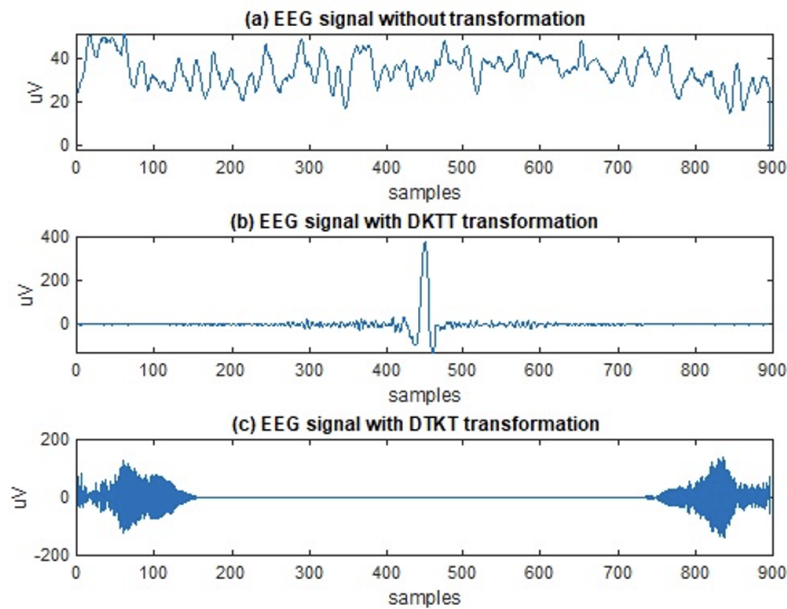


Figure 1: The representation of EEG signal (a) without transformation, (b) with DKTT transformation, and (c) with DTKT transformation

Now the mathematical model of the KTP is represented as follows. The KTP's n th order, $R_n(x)$, is given by [10]:

$$R_n(x) = \sum_{i=0}^{N-1} k_i(n; p, N-1) t_i(x) \quad (1)$$

$$n, x = 0, 1, \dots, N-1$$

$$N > 0, p \in (0, 1)$$

where $t_i(x)$ is the i th order TP given by [8]:

$$t_i(x) = \frac{(1-N)_{i3} F_2(-i, -x, 1+i; 1, 1, 1-N; 1)}{\sqrt{(2i)! \binom{N+i}{2i+1}}} \quad (2)$$

$$i, x = 0, 1, \dots, N-1, N > 0$$

where $\binom{a}{b}$ and $(a)_k$ are the binomial coefficients and the ascending factorial, respectively. $(a)_k$ is defined as follows:

$$(a)_k = a(a+1)(a+2)\dots(a+k-1) = \frac{\Gamma(a+k)}{\Gamma(a)} \quad (3)$$

$k_i(n; p, N-1)$ is the KP's i^{th} order given as [15]:

$$k_i(n; p, N-1) = \sqrt{\frac{A}{B}} {}_2F_1(-i, -n; -N+1; 1/p)$$

$$A = \binom{N-1}{n} p^n (1-p)^{N-1-n}$$

$$B = (-1)^i \left(\frac{1-p}{p}\right)^i \frac{i!}{(-N+1)_i}$$

$$i, x = 0, 1, \dots, N-1,$$

$$N > 0, p \in (0, 1)$$
(4)

At last, ${}_3F_2$ and ${}_2F_1$ are the hypergeometric functions given by [10]:

$${}_3F_2(-i, -x, 1+i; 1, 1, 1-N; 1) = \sum_{j=0}^{\infty} \frac{(-i)_j (-x)_j (1+i)_j}{(i)_j (1-N)_j j!} \quad (5)$$

$${}_2F_1\left(-i, -n; -N+1; \frac{1}{p}\right) = \sum_{j=0}^{\infty} \frac{(-i)_j (-n)_j \left(\frac{1}{p}\right)_j}{(-N+1)_j j!} \quad (6)$$

B. Sparse Filtering

Instead of focusing on learning the input data distribution, the sparse filtering (SF) algorithm, which is a two-layer unsupervised feature learning algorithm, optimizes the sparsity of the learned representation. It can effectively scale with the input data size [12] and is used in this way frequently. The number of features that ensures optimal solution convergence is the only parameter that needs tuning, making it very easy to use. It has been successfully implemented in image and phone classification [13] and is computationally efficient. The following characteristics of the feature distribution matrix are exploited to optimize sparsity.

- Population sparsity occurs when a small set of activated features only represents each instance.
- Lifetime sparsity means that each feature is enabled for only a few instances.
- High dispersal: All the features must have a uniform contribution.

The SF uses a nonlinear transformation to map input features to as few output features as possible, allowing optimal performance. By averaging the squared values in the feature matrix across all instances, we can achieve a high dispersal by considering the mean squared activations of each feature (resulting in nearly equal values). The averaging ensures sparsity over a lifetime and population sparsity [19].

For an offered training set $\{Y^i\}_{i=1}^M$, where $Y^i \in R^{N \times L}$ is a sample value of the M^{th} sample and $N \ll L$. For this purpose, it is conventional to think of the features as rows in the matrix and the samples as columns; specifically, N is the number of original features, while L is the number of samples. The sparse filtering model is trained with the training set to obtain the weight matrix $W \in R^{N \times P}$. When using sparse filtering, samples are mapped onto their corresponding features $Z^i \in R^{P \times L}$ using the weight matrix, W obtained from training. P is the number of learned features obtained from the original features N where $P \ll N$. Linear features are calculated from each sample using Eq. (7).

$$Z_l^i = W_l^T Y^i \quad (7)$$

where Z_l^i represents the l^{th} feature of the i^{th} sample.

When a function is optimized with sparse filtering, the ℓ_2 norm of the function is used as a criterion. A feature matrix can be constructed using the Z_l^i . For features to be stable on the unit ℓ_2 , the feature distribution matrix must be normalized row by row and column by column. The row and column matrix normalization are shown in Eq. (8) and (9), respectively.

$$\tilde{Z}_l = Z_l / \|Z_l\|_2 \quad (8)$$

$$\hat{Z}^i = \tilde{Z}^i / \|\tilde{Z}^i\|_2 \quad (9)$$

Lastly, the weight matrix solution W is calculated using cost function optimization, as shown in Eq. (10).

$$\sum_{i=1}^M \|\hat{Z}^i\|_1 \quad (10)$$

The term $\|\hat{Z}^i\|_1$ measures the sparsity of the i^{th} sample [13].

Since the feature space \hat{Z}^i is limited to a ℓ_2 -ball, the cost function decreases in sparse feature cases. Thus, population sparsity is a target of the objective function. High dispersion is satisfied by normalized features because all features are kept distinct across all samples according to their ℓ_2 -norm. Lifetime sparsity is accomplished by optimizing both population sparsity and high dispersal. To put it another way, when a population is sparse, there are a lot of blank cells in the feature distribution matrix, and when dispersion is high, those blank cells are spread out evenly. As a result, all characteristics are sparse for a lifetime, as they all contain numerous zero values. Sparse filtering is further explained in [19].

C. Support Vector Machine

In machine learning, classification is a common task. This research employed an SVM classifier to classify the EEG data. Several SVM techniques exist, such as linear, quadratic, and cubic SVM classifiers. SVM is a binary classifier to classify numeric EEG data between two classes. Assume there are two data classes, and the purpose is to determine which class a new data item belongs to. By giving a set of training samples, an SVM algorithm creates a model that assigns each sample to one of two categories, ensuring that a wide gap separates samples from different categories. Then, new samples are predicted based on which side they fall on [20].

In support vector machines, a data point is represented as an n-dimensional vector. Linear SVM uses an (n-1)-dimensional hyperplane to make the distinction. To maximize the difference between the two types of processed data, we can pick two parallel hyperplanes that separate them as much as possible. We call the area bordered by these two hyperplanes the margin. Different hyperplanes can be used to partition the data, but the best one is the one that creates the largest gap between the two groups (see Fig. 2). In other words, the hyperplane is chosen to determine the greatest possible distance between it and the nearest data point on either side. The samples on the border are called support vectors [14] once the maximum-margin hyperplane has been identified.

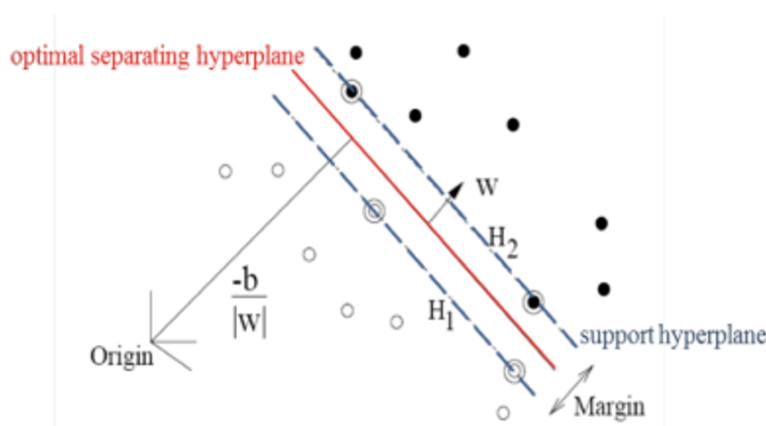


Figure 2: The support vectors and the best separation hyperplane [14]

III. PROPOSED EEG SIGNAL CLASSIFIER

This paper proposes a new method to classify the EEG signal by converting the signal to moments using OP. Moments are passed to a sparse filter to select the most appropriate moments. Finally, the SVM classifier selects the class this signal belongs. Fig. 3 shows the block diagram of the proposed method, and each block is explained below.

1) Training Phase:

- a) Signal preparation: in this block, an EEG signal with M samples sampled at 128 samples per second from one or more channels is arranged in array X of size $N \times M$, where N is the number of training signals.
- b) The classes for each EEG signal are arranged in class table Y of size $N \times 1$.
- c) OP named P of size $M \times M$ is generated for one time using equations (1-6).
- d) Each row of array X is multiplied by the OP P to generate moments with a size of $1 \times M$, arranged in another array Z of size $N \times M$.
- e) The Z array is fed into a pre-trained sparse filter to select the most F -appropriate moments from each row of the Z array and generate moments table D of size $N \times F$.
- f) The moments' table D and the classes' table Y are used to train the SVM classifier.

2) Testing Phase:

- Signal preparation: an EEG signal with M samples sampled at 128 (samples per second) is stored in table X_{test} of size $1 \times M$.
- Multiply the X_{test} with the pre-prepared OP P to generate moments with a size of $1 \times M$ Z_{test} .
- The Z_{test} table is fed into a pre-trained sparse filter to select the most F -appropriate moments from the Z_{test} table and generate moments table D_{test} of size $1 \times F$.
- The SVM classifier uses the moments' table D_{test} to select which class this signal belongs to.

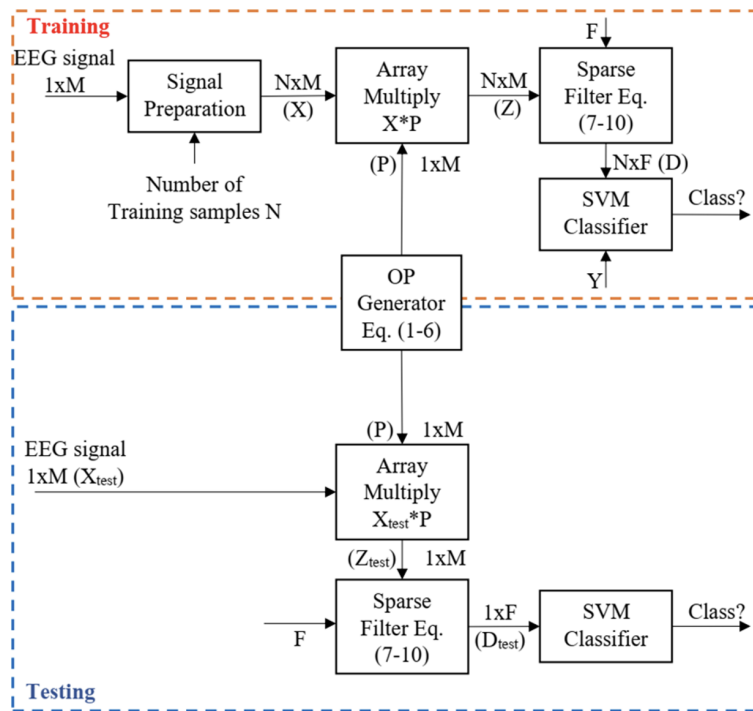


Figure 3: Proposed EEG Signal Classifier Block Diagram

IV. RESULTS AND DISCUSSION

The proposed method's performance is tested by calculating the accuracy of the EEG signal classification. Datasets in [4] and [21] are used to test the proposed method's performance. The first dataset from [4] consists of 20 trials of EEG signal recordings chosen randomly between two classes (left- and right-hand movement). Each trial records 5 seconds of EEG signal at a rate of 128 samples per second, meaning that each trial has 640 samples. In this paper [4], the author uses Linear Discriminant Analysis (LDA) as a features extraction algorithm and hyperplane technique for classification. The second dataset from the globally known dataset "BCI Competition II (dataset number III)" [21] consists of 280 trials of EEG signal recordings chosen randomly between two classes (left- and right-hand movement). Each trial records 6 seconds of EEG signal at a rate of 128 samples per second, meaning that each trial has 768 samples. In this paper [21], Continuous Wavelet Transform (CWT) is applied to the input (EEG) signal to extract the features. After transformation, those features

are classified by the Convolutional Neural Network (CNN), which has one convolution layer, one max-pooling layer, and one fully connected layer. The EEG samples are divided into training samples (80%) and testing samples (20%). 5 - fold cross-validation is used to evaluate the proposed system in classification accuracy. The classification accuracy of the proposed method and the other two methods are shown in Table I.

TABLE I
 COMPARISON OF PROPOSED EEG CLASSIFIER WITH CLASSIFIERS IN [4] AND [21].

Methods	Accuracy (%)		
	Worst	Best	Average
Dataset in [4]			
LDA [4]	60.0	90.0	75.0
Proposed EEG classifier	85.8	95.6	90.7
Dataset in [21]			
Transform-Based CNN [21]	85.7	94.6	90.0
Proposed EEG classifier	82.4	99.5	91.0

The results in Table I show that the proposed method has the highest accuracy among other methods used in the comparison. The conversion of the EEG signal to moments using OP and then reducing the number of those moments by the sparse filter will increase the SVM classifier accuracy. The sparse filter used in the proposed method plays an important role in the performance of this method. Table II will demonstrate this role by showing how the classification accuracy is affected by the change in the number of moments selected.

TABLE II
 EFFECT OF NUMBER OF MOMENTS SELECTED BY THE SPARSE FILTER IN THE CLASSIFIER ACCURACY

No. of moments (F)	Accuracy (%)
100	74.5
150	85.2
200	90.5
250	93.8
300	94.3
350	99.5
400	97.1
450	96.5

Table II shows that the classification accuracy became over 90% when the number of moments selected is about 30% of the original EEG signal samples.

V. CONCLUSION

The paper aimed to design an EEG signal classification method that used OP to transform the spatial EEG signal to moments and then reduce the number of those moments using a sparse filter. The results of this method applied to available EEG datasets of two classes show that the proposed method outperformed other methods by giving an accuracy of over 99% in its best case. The method shows that when the number of moments the sparse filter selects exceeds 30%, the

accuracy will exceed 90%.

Funding

None

ACKNOWLEDGEMENT

The author would like to thank the reviewers for their valuable contribution in the publication of this paper.

CONFLICTS OF INTEREST

The author declares no conflict of interest.

REFERENCES

- [1] R. Abiri, S. Borhani, E. W. Sellers, Y. Jiang, and X. Zhao, "A comprehensive review of EEG-based brain-computer interface paradigms," *J. Neural Eng.*, vol. 16, no. 1, p. 011001, 2019.
- [2] C. Ieracitano, N. Mammone, A. Hussain, and F. C. Morabito, "A novel explainable machine learning approach for EEG-based brain-computer interface systems," *Neural Comput Appl*, vol. 34, no. 14, pp. 11347–11360, 2022.
- [3] Z. Jin, G. Zhou, D. Gao, and Y. Zhang, "EEG classification using sparse Bayesian extreme learning machine for brain-computer interface," *Neural Comput Appl*, vol. 32, no. 11, pp. 6601–6609, 2020.
- [4] M. Z. Al-Faiz and H. A. Abdulkareem, "OFFLINE LINEAR DISCRIMINANT ANALYSIS CLASSIFICATION OF TWO CLASS EEG SIGNALS," *Iraqi Journal of Information and Communications Technology(IJICT)*, vol. 2, no. 3, 2019, doi: <https://doi.org/10.31987/ijict.2.3.71>.
- [5] K. Venkatachalam, A. Devipriya, J. Maniraj, M. Sivaram, A. Ambikapathy, and S. A. Irai, "A Novel Method of motor imagery classification using EEG signal," *Artif Intell Med*, vol. 103, p. 101787, 2020.
- [6] M. Z. Al-Faiz and A. A. Al-Hamadani, "ANALYSIS AND IMPLEMENTATION OF BRAIN WAVES FEATURE EXTRACTION AND CLASSIFICATION TO CONTROL ROBOTIC HAND," *Iraqi Journal of Information and Communications Technology(IJICT)*, vol. 1, 2018, doi: <https://doi.org/10.31987/ijict.1.3.35>.
- [7] H. Karmouni, T. Jahid, M. Savvouri, A. Hmimid, A. El-Affar, and H. Qjidaa, "Image Analysis by Hahn Moments and a Digital Filter," in *Advanced Intelligent Systems for Sustainable Development (AI2SDâ2018)*, M. Ezziyyani, Ed., Cham: Springer International Publishing, 2019, pp. 707–718.
- [8] H. Wu and S. Yan, "Computing invariants of Tchebichef moments for shape-based image retrieval," *Neurocomputing*, vol. 215, pp. 110–117, 2016, doi: <https://doi.org/10.1016/j.neucom.2015.05.147>.
- [9] W. A. Jassim, P. Raveendran, and R. Mukundan, "New orthogonal polynomials for speech signal and image processing," *IET Signal Processing*, vol. 6, no. 8, pp. 713–723(10), Oct. 2012, [Online]. Available: <https://digital-library.theiet.org/content/journals/10.1049/iet-spr.2011.0004>
- [10] B. M. Mahmmod, A. R. bin Ramli, S. H. Abdhussain, S. A. R. Al-Haddad, and W. A. Jassim, "Signal compression and enhancement using a new orthogonal-polynomial-based discrete transform," *IET Signal Processing*, vol. 12, no. 1, pp. 129–142, Feb. 2018, doi: <https://doi.org/10.1049/iet-spr.2016.0449>.
- [11] S. H. Abdhussain, A. R. Ramli, B. M. Mahmmod, M. I. Saripan, S. A. R. Al-Haddad, and W. A. Jassim, "A New Hybrid form of Krawtchouk and Tchebichef Polynomials: Design and Application," *J Math Imaging Vis*, vol. 61, no. 4, pp. 555–570, 2019, doi: 10.1007/s10851-018-0863-4.
- [12] S. Li, J. Wang, X. Jiang, and C. Cheng, "An automatic feature extraction method and its application in fault diagnosis," *Journal of Vibroengineering*, vol. 19, no. 4, pp. 2521–2533, 2017, doi: 10.21595/jve.2017.17906.
- [13] F. M. Fenner and K. Chen, "Towards Understanding Sparse Filtering: A Theoretical Perspective," *CoRR*, vol. abs/1603.08831, 2016, [Online]. Available: <http://arxiv.org/abs/1603.08831>
- [14] S. A. Abdulrahman, M. Roushdy, and A.-B. M. Salem, "SUPPORT VECTOR MACHINE APPROACH FOR HUMAN IDENTIFICATION BASED ON EEG SIGNALS," *JOURNAL OF MECHANICS OF CONTINUA AND MATHEMATICAL SCIENCES*, vol. 15, no. 2, Feb. 2020, doi: 10.26782/jmcms.2020.02.00023.
- [15] D. Rivero-Castillo, H. Piieira, and P. Assuncao, "Edge detection based on Krawtchouk polynomials," *J. Comput. Appl. Math.*, vol. 284, pp. 244–250, 2015, doi: <https://doi.org/10.1016/j.cam.2014.11.017>.
- [16] S. H. Abdhussain, A. R. Ramli, S. A. Al-Haddad, B. M. Mahmmod, and W. A. Jassim, "On Computational Aspects of Tchebichef Polynomials for Higher Polynomial Order," *IEEE Access*, vol. 5, pp. 2470–2478, 2017, doi: 10.1109/ACCESS.2017.2669218.
- [17] G. Pal, D. Rudrapaul, S. Acharjee, R. Ray, S. Chakraborty, and N. Dey, "Video Shot Boundary Detection: A Review," in *Emerging ICT for Bridging the Future - Proceedings of the 49th Annual Convention of the Computer Society of India CSI Volume 2*, S. C. Satapathy, A. Govardhan, K. S. Raju, and J. K. Mandal, Eds., Cham: Springer International Publishing, 2015, pp. 119–127.
- [18] A. C. Den Brinker, "Stable calculation of Tchebichef moments from triplet relations," *Mathematics*, vol. 9, no. 16, Aug. 2021, doi: 10.3390/math9161972.
- [19] J. Ngiam, Z. Chen, S. Bhaskar, P. Koh, and A. Ng, "Sparse Filtering," in *Advances in Neural Information Processing Systems*, J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K. Q. Weinberger, Eds., Curran Associates, Inc., 2011. [Online]. Available: <https://proceedings.neurips.cc/paper/2011/file/192fc044e7d4ffea144af9ac5dc9f3395-Paper.pdf>
- [20] M. A. Hearst, S. T. Dumais, E. Osuna, J. Platt, and B. Scholkopf, "Support vector machines," *IEEE Intelligent Systems and their applications*, vol. 13, no. 4, pp. 18–28, 1998.
- [21] J. Kim, Y. Park, and W. Chung, "Transform-based feature construction utilizing magnitude and phase for the convolutional neural network in EEG signal classification," in *2020 8th International Winter Conference on Brain-Computer Interface (BCI)*, IEEE, Feb. 2020, pp. 1–4, doi: 10.1109/BCI48061.2020.9061635.