Afyon Kocatepe Üniversitesi Fen ve Mühendislik Bilimleri Dergisi
Afyon Kocatepe University Journal of Science and Engineering

AKÜ FEMÜBİD **XX** (202X) xxxxxx (xx s) AKU J. Sci. Eng. **XX** (202X) xxxxxx (xx pp)
DOI:

Araştırma Makalesi / Research Article

Novel Lorentzian Radial Basis Function Kernel

Mustafa Köse[1]
[1] *İstanbul Arel Üniversitesi, Mühendislik ve Mimarlık Fakültesi, Bilgisayar Mühendisliği Bölümü, 541 423 28 25*

Geliş Tarihi: ; Kabul Tarihi:

Keywords
Machine learning; Radial Basis Function Kernel; Lorentzian Metric; Support Vector Machine; Classification

Abstract

In this study we introduce a novel Lorentzian Radial Basis Function(RBF) kernel for use in Support Vector Machine classification. Traditional linear and non-linear methods in machine learning can be considered as decent enough but often they struggle. Our approach leverages the Lorentzian Metric from Lorentzian Space, a concept in differential geometry and mathematical physics, to enhance kernel-based learning methods. Then we perform hyperbolic rotation known as Lorentz Transform, followed by dimension reduction using PCA. And at last, classification using our custom Lorentzian-RBF kernel. The results demonstrate improved performance in some datasets but worsen performance in others. Especially in the digit dataset, an image dataset, which is due to dimension reduction using PCA.

1. Introduction

Traditionally, theory and algorithms of machine learning and statistics have been very well developed for the linear case. Real world data analysis problems, on the other hand, often require nonlinear methods to detect the kind of dependencies that allow successful prediction of properties of interest. By using a positive definite kernel, one can sometimes have the best of both worlds. (Hoffmann et al., 2008)

The strength of kernel methods lies in their flexibility and ability to adapt to different types of data geometries and distributions. This adaptability is crucial in machine learning, where the nature of data can vary greatly across different applications. Kernel methods are a class of effective pattern recognition algorithms that are well suited to model nonlinear relations between the response and predictors. (Matabuena et al., 2022)
Kernel methods can learn nonlinear problems by making an implicit transformation from a low-dimensional input space into a higher-dimensional feature space. (Dral, 2022) And thus allowing linear methods to be applied to non-linear problems. This ability to capture complex patterns and relationships in data has made kernel methods a cornerstone in the advancement of machine learning techniques.

Among the most widely used kernels are the Radial Basis Function (RBF), linear kernels and polynomial kernels. The RBF kernel is a popular kernel function that is used in pattern analysis to precisely measure the Euclidean distance between x and z

$$K_{RBF}(x,\ z)\ =\ exp(\frac{-||x-z||^2}{2\sigma^2})$$

where σ is a kernel parameter and can be recognized as the “spread” of the kernel. (Ding et al., 2021)
Polynomial kernel generates new features by applying a polynomial combination of existing features.

$$K_{poly}(x,\ z)\ =\ (xz\ +\ C)^d$$

d parameter declares the degree of the polynomial kernel. (Rochim et al., 2021)
The most basic of all is the linear kernel, whose expression can be found below

$$K_{linear}(x,\ z)\ =\ (x)^T z$$

Essentially, the linear kernel evaluates the similarity of data points in the original input space using a dot product. (Martínez, 2023)

As of late, attempts to boost the representational power of structured-data by generalizing the kernel methods to non-linear geometries have gained increasing attention. The common strategy to define a valid pd kernel on non-Euclidean geometries is to adopt a proper distance metric. (Fang et al., 2021)

These distances can be listed such as, Graph Distance, where you find the similarity of graphs or the shortest path between two nodes in a graph (Kriege et al., 2020), Cosine Similarity, which calculates the similarity, especially used in text analysis, where the text is mapped into a vector space model, and the similarities between these vectors are calculated as (Qurashi et al., 2020)

$$cos(S_1, S_2) = \frac{S_1 S_2}{||S_1|| ||S_2||}$$

And, the concept that we are going to use in this study, Lorentzian Metric, is a property in Lorentzian Space. Lorentzian Space is an active field of mathematical research that can be seen as part of differential geometry as well as mathematical physics. It represents the mathematical foundation of general relativity which is probably one of the most successful and beautiful theories of physics. (Liu et al., 2010, #)

In this study, we will first apply the Lorentz Transformation to perform hyperbolic rotations on the data, then we will use standard PCA for dimension reduction and lastly we will use SVM with a novel custom kernel based on Lorentzian Metric Tensor for classification.

1.1 Related Work

Even though standardized kernels we use today such as RBF, Linear and Poly are considered sufficient enough, there is always some room for improvements and new experiments. One such custom kernel example developed by Sergei Manzhos et al., focuses on a novel Gaussian Process Regression (GPR) kernel design. This kernel is used for representing multivariate functions with low-dimensional terms. It simplifies and accelerates the machine learning process for fitting potential energy surfaces and kinetic energy densities from sparse data. (Manzhos et al., 2022), in another example Zoran Krunic et al. empirically demonstrated that custom quantum kernels, computed on an IBM quantum computer, could provide an advantage over classical radial basis function kernels in specific configurations of feature sets and training samples. (Krunic et al., 2022)

G. Dheepak et al. have implemented Morlet Wavelet in their custom Wavelet Kernel. A wavelet kernel is grounded in the wavelet transform, a mathematical technique for signal processing, which facilitates the interpretation of data at multiple resolutions. (Dheepak et al., 2023)

And, as for the Lorentzian-based Kernels, Xu et al. leverages the Lorentzian Distance to develop LorentzFM, a model that embeds feature interactions in hyperbolic space. This approach, utilizing the unique properties of Lorentzian geometry, significantly improves the efficiency and performance of recommendation systems and click-through rate predictions, achieving better results with fewer parameters compared to traditional models. (Xu & Wu, 2020)

Another study by Kerimbekov et al. proposes a novel classification method named Lorentzian Distance Classifier for Multiple Features (LDCMF). This method is based on the Lorentzian metric from Lorentzian space and is adapted for datasets with multiple features. The researchers introduced a new Feature Selection in Lorentzian Space (FSLS) method, which selects significant feature pair subsets using a discriminative criterion based on the Lorentzian metric. Their method also includes a data preprocessing step to make the data suitable for Lorentzian space. (Kerimbekov & Bilge, 2017)

2. Material and Method

2.1 Lorentzian Metric Tensor based RBF

The RBF kernel is a kernel family where distance measurements are smoothed by radials function, (Suo et al., 2008, #) And the kernel can be written as below

$$k_{rbf-e}(x, y) = exp(- \gamma(D_E(x, y)^2) \quad (1)$$

where $D_E(x, y)$ refers to distance between x and y points in Euclidean manner and calculated as

$$D_E(x, y) = \sqrt{(x_2 - x_1) + (y_2 - y_1)} \quad (2)$$

and thus the standard RBF Kernel is

$$k_{rbf-e}(x, y) = exp(- \gamma((x_2 - x_1) + (y_2 - y_1))) \quad (3)$$

Now, for our novel Lorentzian Metric based kernel, which will take in place of the Euclidean Distance, we first need to introduce the Lorentzian Metric Tensor and implement a distance calculating method using the metric tensor.
The metric matrix in Lorentzian Space by Liu et al. can be shown as

$$G = \begin{bmatrix} \Lambda_{(n-1)x(n-1)} & 0 \\ 0 & -\lambda \end{bmatrix} \quad (4)$$

$\Lambda_{(n-1)x(n-1)}$ is a diagonal and its diagonal entries are and λ is positive.

Suppose that $r = \left[\hat{r}^T \check{r}\right]^T$ is an n-dimensional vector(a row in a dataset) then a metric tensor g(r, r) with respect to G is expressible as

$$g(r, r) = r^T Gr = \hat{r}^T \Lambda \hat{r} - \lambda(\check{r})^2 \quad (5)$$

and we need to define the distance function using the (5) metric tensor.
x and y to represent two points in the space, we calculate the Lorentzian Distance

$$D_L(x, y) = \sqrt{|g(y - x)|} \quad (6)$$

$$D_L(x, y) = \sqrt{\left|(y - x)^T G(y - x)\right|} \quad (7)$$

Now we have the novel Lorentzian Distance function, before moving on and finalizing our custom RBF kernel, let's take a look at the difference between Euclidean and Lorentzian distances.

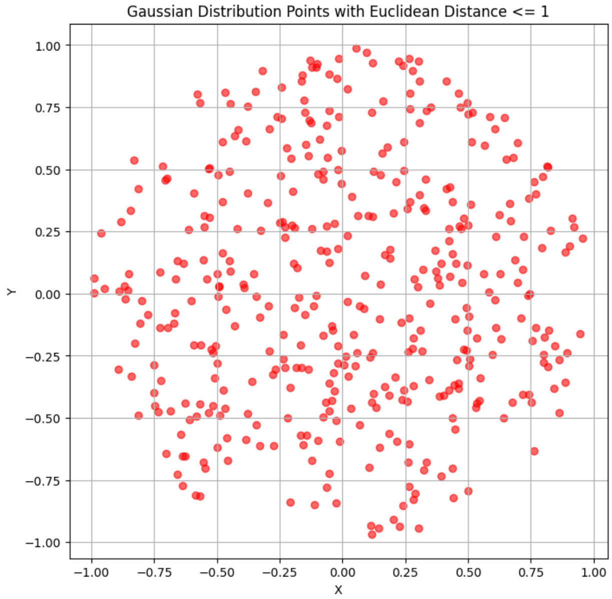

Figure 1. Euclidean Distance of Gaussian Distributed points

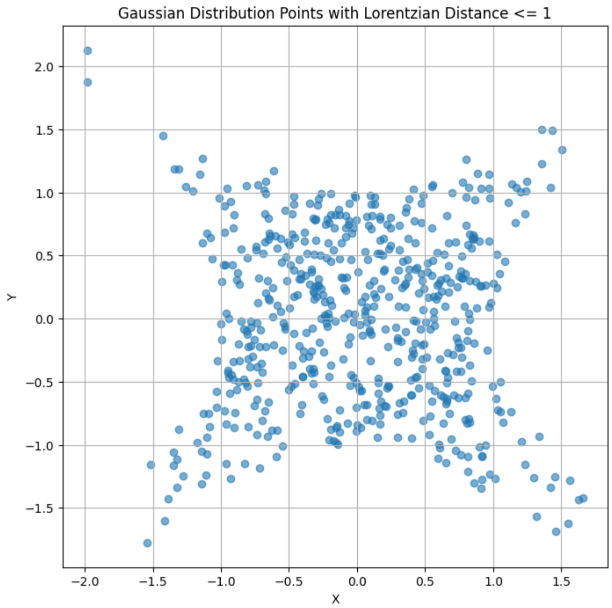

Figure 2. Lorentzian Distance of Gaussian Distributed points

The Euclidean distance is the conventional method of determining distance, analogous to what we commonly use to measure linear distances. In this plot, the points form a circular region around the origin, representing all points that are within a radius of 1 unit from the center. This circular pattern is characteristic of the Euclidean distance in two-dimensional space. Lorentzian on the other hand, the shape and distribution of points in this plot are influenced by its unique metric.

And now, let's define our Lorentzian Distance-RBF

$$k_{rbf-l}(x,\ y)\ =\ exp(-\ \gamma(y\ -\ x)^{T}G(y\ -\ x))\ (8)$$

In our SVM, we are going to use this as a kernel.

2.2 Lorentzian Transform

Recently, there has been interest in using hyperbolic spaces (which are closely related to Lorentzian geometry) in machine learning, particularly for tasks involving complex hierarchical structures or networks. Hyperbolic spaces can represent tree-like structures more efficiently and with less distortion than Euclidean spaces.
And to transform the data into a hyperbolic state we are going to use the Pure Lorentz Boost which is expressed as below (Jaffe, n.d.)

$$K_{L-Transform}(\alpha)\ =\ \begin{bmatrix} cosh(\alpha) & sinh(\alpha) \\ sinh(\alpha) & cosh(\alpha) \end{bmatrix}(9)$$

The α value is also can be optimized for each dataset individually, but in our work, we found that $\alpha\ =\frac{\pi}{2}$ and $\alpha\ =\frac{\pi}{9}$ values to produce the most consistent accuracies.

If X is our data, then the transformed data $X^{'}$ can be obtained by simple dot product

$$X^{'}\ =\ XK_{L-Transform}\ (10)$$

As you can see, to be able to do the dot product, we need 2 dimension data, thus we are going to use PCA for dimension reduction to two features.
The result of the transformation on randomly scattered synthetic data can be seen below

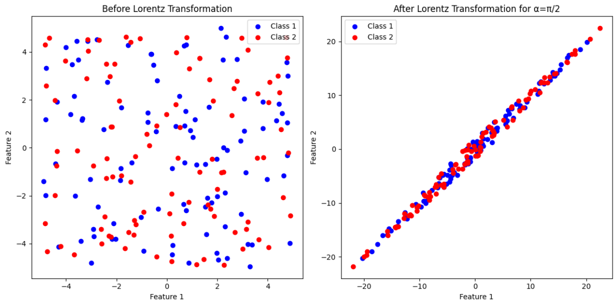

Figure 3. Before and after Lorentzian Transform applied, states of (2 features, 2 classes) data

2.3 Support Vector Machine

Support Vector Machine (SVM) is based on Vapnik Chervonenkis (VC) dimension of statistical learning theory and Structural Risk Minimization.
The main idea of SVM is mapping the non-linear inseparable data into a linear high dimensional feature space. (Song et al., 2008)
The SVM searches for the optimal hyper-plane that has a maximum margin between the nearest positive and negative samples.
This search is expressed as:

$$arg\ min_{w,b}\frac{1}{2}\left\|w^{2}\right\|\ (11)$$

The introduction of the Lagrange multipliers $\alpha\ =\ \{\alpha_i\},\ i\ =\ 1,\ 2,\ ..,\ N$, converts the problem (11) into a maximization problem with respect to α.

$$max_{\alpha}\ D_{\gamma}(\alpha)\ =\ \sum_{i=1}^{N}\alpha_i\ -\frac{1}{2}\sum_{i=1}^{N}\sum_{i=1}^{N}\alpha_i\alpha_j y_i y_j k_{\gamma}(x_i,\ x_j)$$
(12)

where k_{γ} denotes the Euclidean RBF Kernel.
And finally, if we swap out the k_{γ} with our newly proposed Lorentzian-RBF Kernel that is explained in (8), we would come up with our new SVM with novel Lorentzian-RBF Kernel, expressed as below

$$max_{\alpha}D_{\gamma}(\alpha)=\sum_{i=1}^{N}\alpha_i-\frac{1}{2}\sum_{i=1}^{N}\sum_{i=1}^{N}\alpha_i\alpha_j y_i y_j exp(-\ \gamma(x_j-x_i)^{T}G(x_j-x_i))(x_i,\ x_j)$$
(13)

2.4 Parameters

There are a few parameters in this method that needs to be optimized for each dataset individually. One such parameter is the α value in the hyperbolic transformation that happens in the Lorentz Transform (9).

One another method just as important is the Λ, λ scalar values that are in the Metric Matrix (4).

It's computationally heavy to optimize both parameters on their each own. So, for the simplicity we declare one unified parameter for Λ, λ and the γ parameter in the RBF Kernel (8).

$$Combined\ Value\ \in\ [0.5,\ 2.5]$$

Combined Value has been chosen as a scalar in between 0.5 and 2.5, our multiple optimization attempts showed us that the value is most likely going to land on that range for every dataset.

$$\gamma\ =\ \lambda\ =\ Combined\ Value\ (14)$$

$$\Lambda\ =\ \frac{1}{Combined\ Value}\ (15)$$

The effect of the Combined Value on accuracy can be seen in Figure 4.

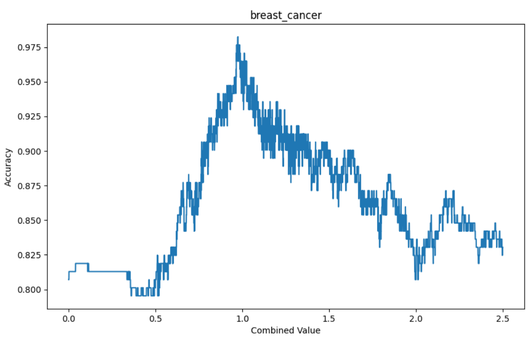

Figure 4. The graph of Combined Value to accuracy of the SVM model for Breast Cancer dataset

2.5 Algorithm

1. Use Standard Scaler from sci-kit library on X to get X’
2. If labels(y) are multi-columns, then we need to convert them into a single column
3. Apply PCA for n_components=2 on X’ to get X’’
4. Apply LorentzTransform to X’’ to get X’’’
5. Use SVM with Lorentzian Distance-RBF Kernel and do classification

Table 1. The algorithm

2.6 Dataset

Alongside the popular sci-kit datasets such as wine, iris, breast cancer etc., we will also focus on one particular dataset. Cardiotograph dataset, sourced from UCI and authored by J. P. Marques de Sá, J. Bernardes, and D. Ayres de Campos, presents a comprehensive collection of 2126 fetal cardiotocograms (CTGs). These CTGs have been processed automatically, allowing for the measurement of various diagnostic features. Importantly, the CTGs have undergone classification by three expert obstetricians, who provided a consensus classification for each record. The classifications are twofold: they describe both a morphologic pattern (denoted by labels A, B, C, etc.) and a fetal state (categorized as N for normal, S for suspect, and P for pathologic). This dual classification system enables the dataset to be versatile, suitable for research purposes in both 10-class and 3-class experimental designs.

The dataset is rich in attributes, providing a detailed insight into the cardiotocographic data. Key attributes include the fetal heart rate (FHR) baseline, number of accelerations, fetal movements, uterine contractions, and various types of decelerations per second (light, severe, prolonged). It also includes metrics like the percentage of time with abnormal short and long-term variability, mean values of short and long-term variability, and various histogram-based measures such as width, minimum, maximum, number of peaks, and zeros. The dataset culminates in key categorical fields: CLASS, which codes for the FHR pattern class (1 to 10), and NSP, which represents the fetal state class code N, S, P(Normal, Suspect, Pathologic). (Campos & Bernardos, 2010)

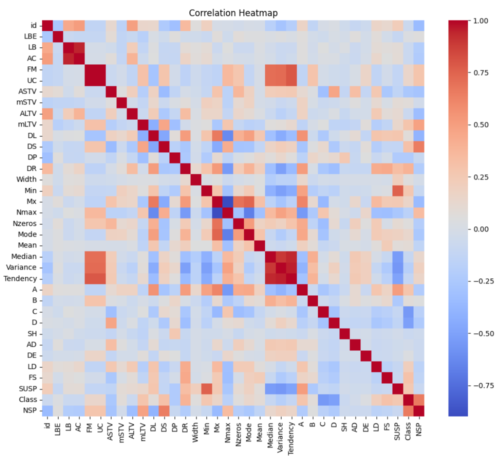

Figure 5. The correlation heatmap of the dataset

As seen in the correlation of features in the dataset, there's a great correlation between labels CLASS and NSP, thus we are going to choose either CLASS or NSP for our classification problem.
Also, there are not many significant positive or negative correlations among many features which gives us one more reason to use PCA for dimension reduction.

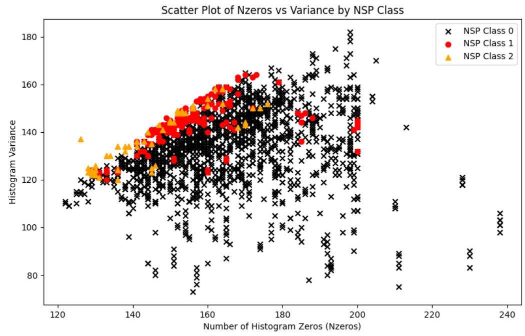

Figure 6. Scatter plot of Nzeros and Variance features

After applying PCA to our dataset, the most two significant features were revealed as Nzeros and Variance features. And so we plot the scatter plot of these features for the classification problem that we chose, NSP.
As seen in the plot above, NSP class 0, which corresponds to the N(Normal) state, seems to dominate in numbers. Although NSP class 1 and class 2, which are S(Suspected) and P(Pathological), are low in numbers, they show great similarity which means there's a great chance that the suspected fetus is most likely a pathological case.

3. Results

As we stated earlier, we are going to test our method in both popular open-source datasets such as Iris, Wine and Breast Cancer as well as the Cardiotocography dataset that was explained in the previous section.
And of course, we are going to compare our algorithm, which its core element is the Lorentzian Distance Kernel-RBF, with other kernels of SVM, like standard RBF, Poly and Linear.

3.1 Wine Dataset

Table 2. SVM classification with different kernels

Kernel Type	α (RBF-L)	Combined Value (RBF-L)	Accuracy
RBF-L(Lorentzian RBF)	$\frac{\pi}{2}$	0.946	1.0
Linear	None	None	0.98148
RBF	None	None	0.98148
Poly	None	None	0.96296

3.2 Iris Dataset

Table 3. SVM classification with different kernels

Kernel Type	α (RBF-L)	Combined Value (RBF-L)	Accuracy
RBF-L(Lorentzian RBF)	$\frac{\pi}{2}$	0.71	0.95555
Linear	None	None	0.97777
RBF	None	None	1.0
Poly	None	None	0.95555

3.3 Breast Cancer Dataset

Table 4. SVM classification with different kernels

Kernel Type	α (RBF-L)	Combined Value (RBF-L)	Accuracy
RBF-L(Lorentzian RBF)	$\frac{\pi}{2}$	0.97449	0.98245
Linear	None	None	0.97660
RBF	None	None	0.97660
Poly	None	None	0.89473

3.4 Digits Dataset

Table 5. SVM classification with different kernels

Kernel Type	α (RBF-L)	Combined Value (RBF-L)	Accuracy
RBF-L(Lorentzian RBF)	$\frac{\pi}{2}$	1.0125	0.58148
Linear	None	None	0.97777
RBF	None	None	0.97962
Poly	None	None	0.95185

3.5 Cardiotocography Dataset

Table 6. SVM classification with different kernels

Kernel Type	α (RBF-L)	Combined Value (RBF-L)	Accuracy
RBF-L(Lorentzian RBF)	$\frac{\pi}{2}$	1.05549	0.90595
Linear	None	None	0.97962
RBF	None	None	0.98432
Poly	None	None	0.9796

3.6 Conclusion

As the results above show, this novel Lorentzian Distance RBF Kernel can deliver high accuracies. With more hyperparameter tuning it might achieve better results on the datasets that it's fallen behind.

And the results for the Digits dataset suggests that this custom model, at least in its current state, is not good for image classification as it is using PCA for dimension reduction; it just makes a good chunk of information non existent. The digits dataset from sci-kit for example, is made of 8x8 handwritten images of digits. So, every image has 8x8=64 features, using PCA for reducing it to only 2 features for sure results in a very bad accuracy. For image classification we need to adapt a different solution for Lorentz Transformation part (9), so we don't need to reduce the dimension of the dataset to 2.
Also, the custom kernel we've developed should be fully compatible with Kernel FDA, Kernel K-Means Clustering and other kernel methods, which would be a great way to confirm that this custom kernel is indeed working as you expect from a kernel.

4. Kaynakça

Campos, D., & Bernardos, J. (2010). Cardiotocography. *UCI Machine Learning*. doi.org/10.24432/C51S4N

Dheepak, G., Christaline, A. J., & Vaishali, D. (2023). MEHW-SVM multi-kernel approach for improved brain tumour classification. *IET Image Processing*, 1-19. doi.org/10.1049/ipr2.12990

Ding, X., Liu, J., Yang, F., & Cao, J. (2021). Random radial basis function kernel-based support vector machine. *Journal of the Franklin Institute*, *358*(18), 10121-10140.

https://doi.org/10.1016/j.jfranklin.2021.10.005

Dral, P. O. (Ed.). (2022). *Quantum Chemistry in the Age of Machine Learning*. Elsevier Science. https://doi.org/10.1016/B978-0-323-90049-2.00009-3

Fang, P., Harandi, M., & Petersson, L. (2021). Kernel Methods in Hyperbolic Spaces. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)* (pp. 10665-10674). IEEE.

Hoffmann, T., Schölkopf, B., & Smola, A. J. (2008). Kernel methods in machine learning. *The Annals of Statistics*, *36*(3), 1171-1220. https://doi.org/10.1214/009053607000000677

Jaffe, A. (n.d.). *Lorentz transformations, rotations, and boosts*. Harvard University.

Kerimbekov, Y., & Bilge, H. Ş. (2017). Lorentzian Distance Classifier for Multiple Features. *6th International Conference on Pattern Recognition Applications and Methods (ICPRAM)*, 493-501. doi.org/10.5220/0006197004930501

Kriege, N. M., Johansson, F. D., & Morris, C. (2020). A survey on graph kernels. *Applied Network Science*, *6*(5). https://doi.org/10.1007/s41109-019-0195-3

Krunic, Z., Flöther, F. F., Seegan, G., Earnest-Noble, N. D., & Shehab, O. (2022). Quantum Kernels for Real-World Predictions Based on Electronic Health Records. *IEEE Transactions on Quantum Engineering*, *1*(11). doi.org/10.1109/TQE.2022.3176806

Liu, R., Lin, Z., Su, Z., & Tang, K. (2010). Feature extraction by learning Lorentzian metric tensor and its extensions. *Pattern Recognition*, *43*(10), 3298–3306. https://doi.org/10.1016/J.PATCOG.2010.05.009

Manzhos, S., Sasaki, E., & Ihara, M. (2022). Easy representation of multivariate functions with low-dimensional terms via Gaussian process regression kernel design: applications to machine learning of potential energy surfaces and kinetic energy densities from sparse data. *Machine Learning: Science and Technology*, *1*(3). doi.org/10.1088/2632-2153/ac4949

Martínez, A. A. (2023, June 26). *Kernel methods with mixed data types and their applications*. UPC, Facultat d'Informàtica de Barcelona, Departament de Ciències de la

Computació. http://hdl.handle.net/2117/394523

Matabuena, M., Félix, P., García-Meixide, C., & Gude, F. (2022). Kernel machine learning methods to handle missing responses with complex predictors. Application in modelling five-year glucose changes using distributional representations. *Computer Methods and Programs in Biomedicine*, (221), 106905. https://doi.org/10.1016/j.cmpb.2022.106905

Qurashi, A. W., Holmes, V., & Johnson, A. P. (2020). Document Processing: Methods for Semantic Text Similarity Analysis. In *2020 International Conference on INnovations in Intelligent SysTems and Applications (INISTA)* (pp. 1-6). IEEE. doi.org/10.1109/INISTA49547.2020.9194665

Rochim, A. F., Widyaningrum, K., & Eridani, D. (2021). Performance Comparison of Support Vector Machine Kernel Functions in Classifying COVID-19 Sentiment. In *2021 4th International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)* (pp. 224-228). IEEE. doi.org/10.1109/ISRITI54043.2021.9702845

Song, H., Ding, Z., Guo, C., Li, Z., & Xia, H. (2008). Research on Combination Kernel Function of Support Vector Machine. *2008 International Conference on Computer Science and Software Engineering*, *1*, 838-841. doi.org/10.1109/CSSE.2008.1231

Suo, H., Li, M., Lu, P., & Yan, Y. (2008). Using SVM as Back-End Classifier for Language Identification. *EURASIP Journal on Audio, Speech, and Music Processing*, *1*. https://asmp-eurasipjournals.springeropen.com/articles/10.1155/2008/674859

Thurnhofer-Hemsi, K., López-Rubio, E., & Molina-Cabello, M. A. (2020). *RADIAL BASIS FUNCTION KERNEL OPTIMIZATION FOR SUPPORT VECTOR MACHINE CLASSIFIERS*. University of Málaga.

Xu, C., & Wu, M. (2020). Learning Feature Interactions with Lorentzian Factorization Machine. *Proceedings of the AAAI Conference on Artificial Intelligence*, *34*(04), 6470-6477. doi.org/10.1609/aaai.v34i04.6119