Science and Technology Indonesia
e-ISSN:2580-4391 *p*-ISSN:2580-4405
Vol. 10, No. 4, October 2025

Research Paper

The Kernel Function of Reproducing Kernel Hilbert Space and Its Application on Support Vector Machine

Bernadhita Herindri Samodera Utami[1,2*], Warsono[2], Mustofa Usman[2], Fitriani[2]

[1]*Doctoral Student of Mathematics and Natural Science, Faculty of Mathematics and Natural Science, Universitas Lampung, Lampung, 35145, Indonesia*
[2]*Department of Mathematics, Faculty of Mathematics and Natural Science, Universitas Lampung, Lampung, 35145, Indonesia*
***Corresponding author**: bernadhita@fmipa.unila.ac.id

Abstract

Reproducing Kernel Hilbert Space (RKHS) is a Hilbert space consisting of functions that can be represented or reproduced by a kernel function. The development of data science has made RKHS a method that refers to an approach or technique using the concept of reproducing kernels in certain applications, especially machine learning. Support Vector Machine (SVM) is one of the machine learning methods included in the supervised learning category for classification and regression tasks. This research aims to determine the form of linear kernel functions, polynomial kernel functions, and Gaussian kernel functions in Support Vector Machine analysis and analyze their performance in Support Vector Machine classification and regression. Application of the RKHS method in SVM classification analysis using World Disaster Risk Dataset data published by Institute for International Law of Peace and Armed Conflict (IFHV) from Ruhr-University Bochum in 2022 obtained results that are based on the results by comparing the predictions of training data and testing data using linear kernel functions, polynomial kernels and Gaussian kernels, it is recommended that classification using linear kernels provides the best prediction performance.

Keywords

Linear Kernel Function, Polynomial Kernel Function, Gaussian Kernel Function, Reproducing Kernel Hilbert Space

Received: 19 January 2025, Accepted: 9 July 2025
https://doi.org/10.26554/sti.2025.10.4.1096-1108

1. INTRODUCTION

Reproducing Kernel Hilbert Space (RKHS) is a Hilbert space consisting of functions that can be represented or reproduced by a kernel function (Awan et al., 2020). The main components of the RKHS concept include Hilbert space, reproducing property, and kernel function (Chen et al., 2021; Paulsen and Raghupathi, 2009). Hilbert space has the special property that every Cauchy sequence is convergent and is equipped with an inner product (inner product) to calculate the length of the vector and the distance and angle between vectors (Utami et al., 2023). Reproducing property means that the evaluation of a function at a point can be defined as the inner product between the original function and the kernel function (Yang, 2016).

The contribution of the RKHS method is explained in the research of Paiva et al. (2019), Akgül (2019), and Rosipal and Trejo (2001), which plays a role in the field of neurophysiology, chemical engineering, and non-linear regression models in high-dimensional space. Along with its development, the RKHS method has become a strong theoretical basis for understanding and developing machine learning algorithms involving kernels (Arqub et al., 2013). Applying the RKHS method in machine learning can solve non-linear regression and classification problems in high-dimensional feature spaces (terms for independent variables) without explicitly calculating transformations. The development of machine learning is relatively fast, only in ten years. Briefly, the machine learning era can be grouped into three eras: the era before 1980, the 1980s, and the 1990s to the present (Shukla et al., 2023).

Support Vector Machine (SVM) is a machine learning algorithm that falls under the broader field of artificial intelligence. Machine learning focuses on creating computer systems capable of learning from data, identifying patterns, and making decisions with minimal human involvement (Muthukrishnan et al., 2020). Its primary objective is to design algorithms or models that enable computers to carry out specific tasks autonomously by learning from training data. The SVM method is used for classification analysis and regression analysis, where the model is trained using pairs of input (independent variables) and output (dependent variables) data that have been labeled. After the model is trained, the machine can be used to classify and build regression models from new data that has never been seen before (Pisner and Schnyer, 2020; Rampisela and Rustam, 2018; Anjani et al., 2024). Initially, SVM was used to classify data into only two classes, but its development SVM can be extended to multi-class classification(Cervantes et al., 2020; Azarnavid et al., 2015; Pisner and Schnyer, 2020). This is supported by the research of Mohan et al. (2020), where they have tried pattern recognition using data mining classification techniques. The results showed that the SVM algorithm produced good pattern prediction and low data classification errors. Furthermore, Zhu et al. (2020) proposed the Huberized Pinball Support Vector Machine (HPSVM) method to improve the resilience of SVM to outliers. This motivates this study to examine the application of the RKHS method in SVM.

One of the interesting things about the Support Vector Machine (SVM) algorithm is that it can be applied to analyze text categorization data, face detection, bioinformatics, and even used to study natural disaster areas that include patterns or characteristics that can be identified by SVM Ghosh et al. (2019), Tuo et al. (2023), Muthukrishnan et al. (2020), and Shukla et al. (2023). In this study, the application of the kernel function in the analysis of Support Vector Machine classification and regression using disaster risk area data from 197 countries from 2011 to 2021. Support Vector Machine is believed to be a powerful and effective classification algorithm, especially in handling high-dimensional data (Rampisela and Rustam, 2018), so it is suitable for use in the analysis of disaster risk data from 197 countries from 2011 to 2021, which is panel data. Panel data is a type of data that involves observations made on the same unit over several periods (Baltagi et al., 2012). In panel data context, SVM can be used for classification or regression tasks by considering the time structure, dependencies between observations, kernel choices, and dimensions (Lestari et al., 2024). If the panel data has a time structure, SVM can model the relationship between variables at a particular time based on historical data.

This study aims to construct the properties of RKHS as a special case of Hilbert space containing unit elements; examines the closed properties of kernel functions applicable to RKHS; construct linear kernel functions, polynomial kernel functions, and Gaussian kernel functions on RKHS; implements linear kernel functions, polynomial kernel functions, and Gaussian kernel functions on Support Vector Machine classification and regression; apply and compare the performance of linear kernel functions, polynomial kernel functions, and Gaussian kernel functions on Support Vector Machine classification and regression using a dataset.

Chakraborty (2012) introduces a Bayesian nonlinear multivariate regression model utilizing RKHS for high-dimensional datasets, particularly in near-infrared spectroscopy. The model adeptly handles multiple correlated responses and employs a robust Bayesian support vector regression framework, highlighting the efficacy of RKHS in complex, high-dimensional regression scenarios. Rincón and Ruiz-Medina (2012) propose a classification methodology combining wavelet bases with RKHS theory to analyze gene expression profiles, showcasing the adaptability of RKHS in functional data analysis. Unlike previous studies such as Chakraborty (2012) and Rincón and Ruiz-Medina (2012), which focused primarily on domain-specific applications of RKHS in spectroscopy and gene expression analysis, this study presents a mathematically grounded construction of classical kernel functions in RKHS and applies them systematically to both classification and regression using high-dimensional panel data on disaster risk. This dual theoretical-practical approach, combined with the comparative analysis of kernel performance, offers both novel insights into RKHS theory and practical recommendations for kernel-based learning in real-world, time-dependent data settings.

The linear kernel function corresponds to the standard inner product in Euclidean space. Even a polynomial kernel introduces non-linearity while retaining interpretability, and a Gaussian kernel maps data into an infinite-dimensional space and is universal under certain conditions. Unlike previous studies that focused on the empirical implementation of kernel functions, this study provides a mathematical analysis of kernel properties in RKHS. It systematically compares the performance of kernel functions on SVM using a data set.

There are four points of novelty in this study. First, this study constructs RKHS properties based on Hilbert space equipped with unit elements in Hilbert space. This means that this study presents the development of theoretical properties of RKHS as a special case of Hilbert space equipped with unit elements. Second, this study examines the closed nature of kernel functions in RKHS and aims to provide new insights into the operations applicable to RKHS. Third, this study discusses linear, polynomial, and Gaussian kernel functions and constructs their formulations and properties in RKHS, which have not been widely explored mathematically in previous studies. Fourth, the implementation and comparative analysis of kernel functions in Support Vector Machine (SVM) using a data set. This study applies linear kernel, polynomial, and Gaussian kernel functions to the classification and regression of SVMs. By comparing the performance of different kernels using a dataset, this study provides recommendations for selecting the best kernel according to the problem context. By integrating rigorous mathematical analysis with applied machine learning, this study provides both a theoretical advancement in RKHS and practical insights for disaster data analysis using SVM.

In this paper, section 2 provides the methods or research steps. Section 3 presents the results and discussion, which consist of three main parts: reproducing kernel Hilbert space on a support vector machine, kernel function on a support vector machine, and performance comparison of kernel function on a support vector machine. Section 4 concludes the findings and provides recommendations for future research and practical kernel selection in high-dimensional and temporal data contexts.

2. EXPERIMENTAL SECTION

2.1 Methods

This section describes the steps taken.

1. Provide the remark of Hilbert space with the unit element based on the given definition.
2. Provide examples of feature mapping in RKHS and the remark based on the examples.
3. Provide the proposition of the Gram matrix, which is useful for proving theorems on the properties of kernel functions.
4. Provide the theorems of properties of the kernel function in RKHS.
5. Construct mathematical formulation of linear, polynomial, and Gaussian kernel functions within the RKHS framework on SVM.
6. Utilize the dataset with a Support Vector Machine (SVM) for classification and regression tasks and evaluate the performance of linear, polynomial, and Gaussian kernel functions.

3. RESULTS AND DISCUSSION

3.1 Reproducing Kernel Hilbert Space on Support Vector Machine

Unlike previous studies that focused on the empirical implementation of kernel functions, this study provides a mathematical analysis of kernel properties in RKHS. It systematically compares the performance of kernel functions on SVM using datasets. Thus, the novelty of this study can be described as follows. This study constructs more specific properties of RKHS by utilizing the concept of unit elements in Hilbert space, which has never been discussed by previous researchers. By studying the closed properties of kernel functions, this study provides new insights into the operations applicable to RKHS. This study applies linear kernel, polynomial kernel, and Gaussian kernel functions to SVM classification and regression. By comparing the performance of different kernels using datasets, this study provides recommendations for selecting the best kernel according to the problem context.

Hilbert space is a key component in RKHS. The following is the definition of a Hilbert space with the unit element (Small and McLeish, 1994).

Definition 3.1 *Given a Hilbert space H. Given an element $\mathbf{1}$. An ordered pair $(H, \mathbf{1})$ is called a Hilbert space equipped with a unit element if $\langle \mathbf{1}, \mathbf{1} \rangle = 1$. Thus, the scalar product of $\mathbf{1}$ is an element of H.*

Remark 3.2 *A Hilbert space that induced by inner product, the unit element is not unique.*

Reproducing Kernel Hilbert Space (RKHS) is fundamentally linked to kernel functions, as a specific type of Hilbert space is defined and constructed through these functions. The following is the definition of RKHS as presented by Berlinet and Thomas-Agnan (2004).

Definition 3.3 *Let Φ be any set. A kernel function $\zeta : \Phi \times \Phi \to \mathbb{R}$ is a reproducing kernel Hilbert space on a Hilbert space H if and only if it satisfies the following conditions:*

1. *for every $v \in \Phi$, $\zeta(v, \cdot) \in H$,*
2. *for every $v \in \Phi$ and for every $\varphi \in H$, $\langle \varphi, \zeta(v, \cdot) \rangle = \varphi(v)$ where $\varphi : \Phi \to \mathbb{R}^n$.*

Remark 3.4 *Based on the Definition 3.1, if $\eta : \Phi \to \mathbb{R}$ and $\zeta(v_i, v_j)$ are chosen in a Hilbert space such that $\langle \eta, \zeta(v_i, v_j) \rangle = \eta(v)$ holds, then the Hilbert space in which the inner product holds is called the RKHS.*

Kernel functions in RKHS can perform inner product computations in high-dimensional spaces without explicitly mapping data to that space. The following defines the kernel functions (Bowman and Azzalini, 1997; Shawe-Taylor and Cristianini, 2004; Ghojogh et al., 2021).

Definition 3.5 *Let Φ be any set. A function $\zeta : \Phi \times \Phi \to \mathbb{R}$ is called a kernel function if for every $v, w \in \Phi$ it satisfies*

$$\zeta(v, w) = \langle \eta(v), \eta(w) \rangle \tag{1}$$

where $\eta : \Phi \to \mathbb{R}^n$ is a mapping from Φ to an n-dimensional feature space.

Here are examples of a kernel function.

Example 3.6 *Let Φ be a two-dimensional set of real numbers $\mathbb{R}^2$ and define a mapping $\eta : \mathbb{R}^2 \to \mathbb{R}^3$:*

$$\eta : v = (v_1, v_2) \to \eta(v) = (v_1^2, v_2^2, \sqrt{2}v_1v_2) \in \mathbb{R}^3. \tag{2}$$

Let $v, w \in \Phi$, by Definition 3.5 applies

$$\begin{aligned} \zeta(v, w) &= \langle \eta(v), \eta(w) \rangle \\ &= \langle (v_1^2, v_2^2, \sqrt{2}v_1v_2), (w_1^2, w_2^2, \sqrt{2}w_1w_2) \rangle \\ &= v_1^2w_1^2 + v_2^2w_2^2 + 2v_1v_2w_1w_2 \\ &= (v_1w_1 + v_2w_2)^2 \\ &= \langle v, w \rangle^2. \end{aligned}$$

Thus, the function $\zeta(v, w) = \langle v, w \rangle^2$ is a kernel function corresponding to the feature space $\mathbb{R}^3$.

Example 3.7 *Let $\Phi = \mathbb{R}^2$ and defined a feature mapping $\eta : \mathbb{R}^2 \to \mathbb{R}^4$:*

$$\eta : v = (v_1, v_2) \to \eta(v) = (v_1^2, v_2^2, v_1v_2, v_2v_1) \in \mathbb{R}^4. \tag{3}$$

Let $v, w \in \Phi$, by Definition 3.5 applies

$$\begin{aligned} \zeta(v, w) &= \langle \eta(v), \eta(w) \rangle \\ &= \langle (v_1^2, v_2^2, v_1v_2, v_2v_1), (w_1^2, w_2^2, w_1w_2, w_2w_1) \rangle \\ &= v_1^2w_1^2 + v_2^2w_2^2 + v_1v_2w_1w_2 + v_2v_1w_2w_1 \\ &= (v_1w_1 + v_2w_2)^2 \\ &= \langle v, w \rangle^2. \end{aligned}$$

Thus, the function $\zeta(v, w) = \langle v, w \rangle^2$ is a kernel function corresponding to the feature space $\mathbb{R}^4$.

Remark 3.8 *Based on Example 3.6 and Example 3.7, the kernel function uniquely determines the feature space.*

In other words, the same kernel function can produce different feature mapping dimensions.

Kernel functions can be represented as a Gram matrix because this matrix simplifies calculations in high-dimensional feature spaces without having to directly calculate the coordinates in that space (Shawe-Taylor and Cristianini, 2004; Ghojogh et al., 2021).

Definition 3.9 *Let V be a set containing $\{\varepsilon_1, \varepsilon_2, ..., \varepsilon_n\}$. The Gram matrix is defined as a matrix $\boldsymbol{G}$ of size $l \times l$ with entries $G_{ij} = \langle \varepsilon_i, \varepsilon_j \rangle$.*

Definition 3.10 *Let X be any set. If ζ is a function that evaluates the inner product based on the feature mapping η then the corresponding Gram matrix has entries $G_{ij} = \langle \eta(\varepsilon_i), \eta(\varepsilon_j) \rangle = \zeta(\varepsilon_i, \varepsilon_j)$.*

Definition 3.11 *Let Φ be any set and A be a symmetric matrix. The matrix A is called semi-positive definite if each eigenvalue is non-negative or $v'Av \geq 0$, for each vector $v \in \Phi$.*

Based on Definition 3.9, Definition 3.10, and Definition 3.11, the following propositions are obtained.

Proposition 3.12 *The Gram matrix and the kernel matrix are semi-positive definite matrices.*

Proof. Let $\mathbf{G}$ be a Gram matrix, where

$$G_{ij} = \zeta(\varepsilon_i, \varepsilon_j) = \langle \eta(\varepsilon_i), \eta(\varepsilon_j) \rangle, \tag{4}$$

for $i, j = 1, 2, \ldots, l$, and $\varepsilon_i, \varepsilon_j \in \Phi$.

Let $v \in V$ be any vector. Then:

$$\begin{aligned} v'\mathbf{G}v &= \begin{pmatrix} v_1 & v_2 & \cdots & v_l \end{pmatrix} \\ &\quad \begin{pmatrix} \zeta(\varepsilon_1, \varepsilon_1) & \cdots & \zeta(\varepsilon_1, \varepsilon_l) \\ \zeta(\varepsilon_2, \varepsilon_1) & \cdots & \zeta(\varepsilon_2, \varepsilon_l) \\ \vdots & \vdots & \ddots \\ \zeta(\varepsilon_l, \varepsilon_1) & \cdots & \zeta(\varepsilon_l, \varepsilon_l) \end{pmatrix} \vdots \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_l \end{pmatrix} \\ &= \begin{pmatrix} v_1 & v_2 & \cdots & v_l \end{pmatrix} \\ &\quad \begin{pmatrix} \langle \eta(\varepsilon_1), \eta(\varepsilon_1) \rangle & \cdots & \langle \eta(\varepsilon_1), \eta(\varepsilon_l) \rangle \\ \langle \eta(\varepsilon_2), \eta(\varepsilon_1) \rangle & \cdots & \langle \eta(\varepsilon_2), \eta(\varepsilon_l) \rangle \\ \vdots & \vdots & \ddots \\ \langle \eta(\varepsilon_l), \eta(\varepsilon_1) \rangle & \cdots & \langle \eta(\varepsilon_l), \eta(\varepsilon_l) \rangle \end{pmatrix} \vdots \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_l \end{pmatrix} \\ &= \sum_{i,j=1}^{l} v_i \langle \eta(\varepsilon_i), \eta(\varepsilon_j) \rangle v_j \\ &= \left\langle \sum_{i=1}^{l} v_i \eta(\varepsilon_i), \sum_{j=1}^{l} v_j \eta(\varepsilon_j) \right\rangle \\ &= \left\| \sum_{i=1}^{l} v_i \eta(\varepsilon_i) \right\|^2 \geq 0. \end{aligned}$$

The following properties are obtained about the kernel function.

Theorem 3.13 *Let any set $\Phi \subseteq \mathbb{R}^n$. Define the functions $\Phi \times \Phi \to \mathbb{R}^n$. Let ζ_1 and ζ_2 be any kernel functions. If η is a mapping from $\Phi \to \mathbb{R}^n$, then the following properties hold.*

1. *$\zeta(\varepsilon, \nu) = \zeta_1(\varepsilon, \nu) + \zeta_2(\varepsilon, \nu)$ is a kernel function, for any $\varepsilon, \nu \in \Phi$.*
2. *$\zeta(\varepsilon, \nu) = \alpha\zeta_1(\varepsilon, \nu)$ is a kernel function, for any $\varepsilon, \nu \in \Phi$ and $\alpha \in \mathbb{R}^+$.*
3. *$\zeta(\varepsilon, \nu) = \zeta_1(\varepsilon, \nu)\zeta_2(\varepsilon, \nu)$ is the kernel function, for any $\varepsilon, \nu \in \Phi$.*

Proof.

1. By Proposition 3.12, then for any $v \in \Phi$ it will be shown $v'(\mathbf{G}_1 + \mathbf{G}_2)v \geq 0$.
   Based on Definition 3.5,
   $$\mathbf{G}_1 = \begin{pmatrix} \zeta_1(\varepsilon_1, \varepsilon_1) & \ldots & \zeta_1(\varepsilon_1, \varepsilon_l) \\ \zeta_1(\varepsilon_2, \varepsilon_1) & \ldots & \zeta_1(\varepsilon_2, \varepsilon_l) \\ \ldots & \ldots & \ldots \\ \zeta_1(\varepsilon_l, \varepsilon_1) & \ldots & \zeta_1(\varepsilon_l, \varepsilon_l) \end{pmatrix} = \begin{pmatrix} \langle \eta_1(\varepsilon_1), \eta_1(\varepsilon_1) \rangle & \ldots & \langle \eta_1(\varepsilon_1), \eta_1(\varepsilon_l) \rangle \\ \langle \eta_1(\varepsilon_2), \eta_1(\varepsilon_1) \rangle & \ldots & \langle \eta_1(\varepsilon_2), \eta_1(\varepsilon_l) \rangle \\ \ldots & \ldots & \ldots \\ \langle \eta_1(\varepsilon_l), \eta_1(\varepsilon_1) \rangle & \ldots & \langle \eta_1(\varepsilon_l), \eta_1(\varepsilon_l) \rangle \end{pmatrix}.$$
   Since ζ_1 is a kernel function it means that $v'(\mathbf{G}_1)v \geq 0$.
   Likewise
   $$\mathbf{G}_2 = \begin{pmatrix} \zeta_2(\varepsilon_1, \varepsilon_1) & \ldots & \zeta_2(\varepsilon_1, \varepsilon_l) \\ \zeta_2(\varepsilon_2, \varepsilon_1) & \ldots & \zeta_2(\varepsilon_2, \varepsilon_l) \\ \ldots & \ldots & \ldots \\ \zeta_2(\varepsilon_l, \varepsilon_1) & \ldots & \zeta_2(\varepsilon_l, \varepsilon_l) \end{pmatrix} = \begin{pmatrix} \langle \eta_2(\varepsilon_1), \eta_2(\varepsilon_1) \rangle & \ldots & \langle \eta_2(\varepsilon_1), \eta_2(\varepsilon_l) \rangle \\ \langle \eta_2(\varepsilon_2), \eta_2(\varepsilon_1) \rangle & \ldots & \langle \eta_2(\varepsilon_2), \eta_2(\varepsilon_l) \rangle \\ \ldots & \ldots & \ldots \\ \langle \eta_2(\varepsilon_l), \eta_2(\varepsilon_1) \rangle & \ldots & \langle \eta_2(\varepsilon_l), \eta_2(\varepsilon_l) \rangle \end{pmatrix}.$$
   Since ζ_2 is a kernel function, it means that $v'(\mathbf{G}_2)v \geq 0$.
   Thus, the sum $v'(\mathbf{G}_1)v + v'(\mathbf{G}_2)v = v'(\mathbf{G}_1 + \mathbf{G}_2)v \geq 0$.
   Thus, $\zeta(\varepsilon, \nu) = \zeta_1(\varepsilon, z) + \zeta_2(\varepsilon, \nu)$ is a kernel function.
2. Let ζ_1 kernel function and $a \in \mathbb{R}^+$. It will be shown that $\alpha v'\mathbf{G}_1 v \geq 0$.
   Based on Definition 3.5,
   $$\mathbf{G}_1 = \begin{pmatrix} \zeta_1(\varepsilon_1, \varepsilon_1) & \ldots & \zeta_1(\varepsilon_1, \varepsilon_l) \\ \zeta_1(\varepsilon_2, \varepsilon_1) & \ldots & \zeta_1(\varepsilon_2, \varepsilon_l) \\ \ldots & \ldots & \ldots \\ \zeta_1(\varepsilon_l, \varepsilon_1) & \ldots & \zeta_1(\varepsilon_l, \varepsilon_l) \end{pmatrix} = \begin{pmatrix} \langle \eta_1(\varepsilon_1), \eta_1(\varepsilon_1) \rangle & \ldots & \langle \eta_1(\varepsilon_1), \eta_1(\varepsilon_l) \rangle \\ \langle \eta_1(\varepsilon_2), \eta_1(\varepsilon_1) \rangle & \ldots & \langle \eta_1(\varepsilon_2), \eta_1(\varepsilon_l) \rangle \\ \ldots & \ldots & \ldots \\ \langle \eta_1(\varepsilon_l), \eta_1(\varepsilon_1) \rangle & \ldots & \langle \eta_1(\varepsilon_l), \eta_1(\varepsilon_l) \rangle \end{pmatrix}.$$
   Since ζ_1 is a kernel function, it means that $v'(\mathbf{G}_1)v \geq 0$. Thus, $v'(\alpha\mathbf{G}_1)v = \alpha v'(\mathbf{G}_1)v \geq 0$. So, $\zeta(\varepsilon, \nu) = \alpha\zeta_1$ (ε, ν) is a kernel function.
3. Given ζ_1 kernel function, meaning $v'(\mathbf{G}_1)v \geq 0$ and ζ_2 kernel function meaning $v'(\mathbf{G}_2)v \geq 0$. It will be shown $v'(\mathbf{G}_1\mathbf{G}_2)v \geq 0$. Based on Schur product so that $v'(\mathbf{G}_1)v \circ v'(\mathbf{G}_2)v = v'(\mathbf{G}_1 \circ \mathbf{G}_2)v \geq 0$. So, $\zeta(\varepsilon, \nu) = \zeta_1(\varepsilon, \nu)\zeta_2$

(ε, ν) is a kernel function.

Based on Theorem 3.13, a new kernel function can be formed as follows.

Theorem 3.14 *Let any set $\Phi \subseteq \mathbb{R}^n$ and a function of $\Phi \times \Phi \to \mathbb{R}^n$. If $\zeta_1, \zeta_2, \cdots$, and ζ_n as finite kernel functions and $\alpha_1, \alpha_2, \cdots$, and α_n as any positive real numbers, then the finite linear combination is a kernel function.*

Proof. For $i = 1, 2, \ldots, n$ every kernel function $\zeta_i(\varepsilon, \nu)$ there is a Gram matrix $\mathbf{G}^{(i)}$ such that

$$\mathbf{G}^{(i)}_{jk} = \zeta_i(\varepsilon_j, \varepsilon_k); \quad j, k = 1, 2, \ldots, m \quad \forall \varepsilon_1, \varepsilon_2, \cdots, \varepsilon_m \in \Phi.$$

Since $\zeta_i(\varepsilon, \nu)$ kernel function then every $\mathbf{G}^{(i)}$ holds $v'\mathbf{G}^{(i)}v \geq 0, \forall v \in \Phi$. Gram matrix $\mathbf{G}$ from linear combination $\zeta(\varepsilon, \nu) = \sum_{i=1}^{n} \alpha_i \zeta_i(\varepsilon, \nu)$ is $\mathbf{G} = \sum_{i=1}^{n} \alpha_i \mathbf{G}^{(i)}$ such that $v'\mathbf{G}v = v'(\sum_{i=1}^{n} \alpha_i \mathbf{G}^{(i)})v$. Since $\alpha_i > 0$ and $v'\mathbf{G}^{i}v \geq 0$ then $\sum_{i=1}^{n} \alpha_i (v'\mathbf{G}^{(i)}v) \geq 0$.

3.2 Kernel Function on Support Vector Machine

Proposition 3.15 *If $\zeta(\varepsilon, \nu)$ is a kernel over $\Phi\times\Phi$, with $\varepsilon, \nu \in \Phi$ then $\zeta(\varepsilon, \nu) = \varepsilon \cdot \nu$ is a kernel function.*

Proof.

Based on Definition 3.5, a function is defined

$$\langle \eta(\varepsilon), \eta(\nu) \rangle = \langle \varepsilon, \nu \rangle = \varepsilon \cdot \nu = \sum_{i=1}^{n} \varepsilon_i \nu_i. \tag{5}$$

According to Definition 3.11, for every $\gamma \in \Phi$ holds

$$\sum_{i=1}^{x} \sum_{j=1}^{x} \gamma_i \gamma_j \zeta(\varepsilon_i, \varepsilon_j) = \sum_{i=1}^{x} \sum_{j=1}^{x} \gamma_i \gamma_j (\varepsilon_i \cdot \varepsilon_j) = \left(\sum_{i=1}^{x} \gamma_i \gamma_i \right) \left(\sum_{j=1}^{x} \gamma_j \gamma_j \right).$$

Let $v = \sum_{i=1}^{x} \gamma_i \varepsilon_i$ then the expression above becomes $v \cdot v = \|v\|^2 \geq 0$. So, $\zeta(\varepsilon, \nu) = \varepsilon \cdot \nu$ is a kernel function.

Remark 3.16 *The kernel function $\zeta(\varepsilon, \nu) = \varepsilon \cdot \nu$ is hereinafter called the linear kernel function.*

Proposition 3.17 *For $\varepsilon, \nu \in \Phi$, if $\zeta_1(\varepsilon, \nu)$ is a kernel over $\Phi\times\Phi$ and $p(\varepsilon) = a_0 + a_1\varepsilon + a_2\varepsilon^2 + \cdots + a_n\varepsilon^n$ is a polynomial with positive real coefficients then $\zeta(\varepsilon, \nu) = p(\zeta_1(\varepsilon, \nu))$ is a kernel function.*

Proof. The polynomial $p(\zeta_1(\varepsilon, \nu)$ is explained as

$$\begin{aligned} p(\zeta_1(\varepsilon, \nu)) &= a_0 + a_1\zeta_1(\varepsilon, \nu) + a_2\zeta_1(\varepsilon, \nu)^2 + \cdots \\ &\quad + a_n\zeta_1(\varepsilon, \nu)^n \qquad (6) \\ &= \sum_{i=0}^{n} a_i \zeta_i(\varepsilon, \nu)^i \qquad (7) \end{aligned}$$

Since $a_i \geq 0$ and ζ_1 is a kernel function, then $a_i\zeta_i(\varepsilon, \nu)^i \geq 0, \forall i = 1, 2, \cdots, n$. Based on Theorem 3.13 (i) that the sum of kernel functions produces a kernel function, then $\zeta(\varepsilon, \nu) = p(\zeta_1(\varepsilon, \nu))$ is a kernel function.

Remark 3.18 *The kernel function $\zeta(\varepsilon, \nu) = a_0 + a_1\zeta_1(\varepsilon, \nu) + a_2\zeta_1(\varepsilon, \nu)^2 + \cdots + a_n\zeta_1(\varepsilon, \nu)^n$ is hereinafter called the polynomial kernel function.*

Proposition 3.19 *For $\varepsilon, \nu \in \Phi$, if $\zeta_1(\varepsilon, \nu)$ a kernel over $\Phi \times \Phi$ then $\zeta(\varepsilon, \nu) = \exp(\zeta_1(\varepsilon, \nu))$ is kernel function.*

Proof. The exponential function $\exp(\zeta_1(\varepsilon, \nu))$ is expressed as a Taylor series expansion

$$\sum_{n=0}^{\infty} \frac{\zeta_1(\varepsilon, \nu)^n}{n!}. \tag{8}$$

Since ζ_1 is a kernel function and $n \geq 0$ then every power of the kernel function is a kernel function. Based on Theorem 3.13 (i) that the sum of kernel functions produces a kernel function then $\zeta(\varepsilon, \nu) = \exp(\zeta_1(\varepsilon, \nu))$ is a kernel function.

Proposition 3.20 *If $\varepsilon, \nu \in \Phi$ then $\zeta(\varepsilon, \nu) = \exp(-\frac{\|\varepsilon-\nu\|^2}{2\psi^2})$ is a kernel function.*

Proof. Let a function

$$\zeta(\varepsilon, \nu) = \exp(-\frac{\|\varepsilon - \nu\|^2}{2\psi^2}). \tag{9}$$

Note that

$$\|\varepsilon - \nu\|^2 = \|\varepsilon\|^2 + \|\nu\|^2 - 2\langle \varepsilon, \nu \rangle \tag{10}$$

so that

$$-\frac{\|\varepsilon - \nu\|^2}{2\psi^2} = -\frac{\|\varepsilon\|^2}{2\psi^2} - \frac{\|\nu\|^2}{2\psi^2} + \frac{2\langle \varepsilon, \nu \rangle}{2\psi^2}. \tag{11}$$

Thus,

$$\begin{aligned} \zeta(\varepsilon, \nu) &= \exp\left(-\frac{\|\varepsilon - \nu\|^2}{2\psi^2} \right) \\ &= \exp\left(-\frac{\|\varepsilon\|^2}{2\psi^2} - \frac{\|\nu\|^2}{2\psi^2} + \frac{2\langle \varepsilon, \nu \rangle}{2\psi^2} \right) \\ &= \exp\left(-\frac{\|\varepsilon\|^2}{2\psi^2} \right) \exp\left(-\frac{\|\nu\|^2}{2\psi^2} \right) \exp\left(\frac{2\langle \varepsilon, \nu \rangle}{2\psi^2} \right) \end{aligned}$$

The value of $\exp(-\frac{\|\varepsilon\|^2}{2\psi^2})$ and $\exp(-\frac{\|\nu\|^2}{2\psi^2})$ depends on each of ε and ν so that it does not affect the positive semi-definite property of the kernel matrix (since it is a scalar). Based on Proposition 3.19, $\exp(\frac{2\langle \varepsilon, \nu \rangle}{2\psi^2})$ is a kernel where $\langle \varepsilon, \nu \rangle$ is a linear kernel function. So, $\zeta(\varepsilon, \nu) = \exp(-\frac{\|\varepsilon-\nu\|^2}{2\psi^2})$ is the kernel function.

Remark 3.21 *The kernel function $\zeta(\varepsilon, \nu) = \exp(-\frac{\|\varepsilon-\nu\|^2}{2\psi^2})$ is hereinafter called the Gaussian kernel function or Radial Basis Function.*

Support Vector Machine (SVM) is a machine learning algorithm to identify the optimal hyperplane that maximally separates different data classes within the feature space. The principle of SVM if using the linear kernel function $\zeta(\varepsilon, \nu) = \varepsilon \cdot \nu$ is as follows.

Let a dataset set $\{(x_i, y_i)\}_{i=1}^{n}$, with $x_i \in \mathbb{R}^d$, $y_i \in \{-1, 1\}$ SVM aims to find a hyperplane in the form $f(x) = w \cdot x + b$ where $w \in \mathbb{R}^d$ is the weight vector and $b \in \mathbb{R}$ is the bias. Classification is defined by

$$\text{prediction label} = \text{sign}(f(x)). \tag{12}$$

SVM maximizes the margin between two classes, namely the distance between the hyperplane and the nearest data (support vectors) with the formula:

$$\text{margin} = \frac{2}{\|w\|}. \tag{13}$$

Optimization is done by solving the primal problem

$$\min_{w,b} \frac{1}{2}\|w\|^2 \quad \text{relative to}$$
$$y_i(w \cdot x_i + b) \geq 1, \ \forall i, i = 1, 2, \cdots, n. \tag{14}$$

The objective function $\frac{2}{\|w\|}$ represents the optimization goal, which is to minimize the norm w, which corresponds to the maximum margin between two classes in SVM. The smaller $\|w\|^2$, the larger the separation margin.

The constraint $y_i(w \cdot x_i + b) \geq 1, \ \forall i$ ensures that the training data x_i is correctly classified, i.e., is on the correct side of the margin with a distance of at least 1.

When the data dimension is low or the number of constraints is small, solving the primal problem is more efficient than the dual. However, for SVM problems with high dimensional data, the dual approach is preferred due to its computational simplicity.

The form of the dual problem is

$$\max_{\alpha} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \zeta(x_i, x_j) \tag{15}$$

relative to $\sum_{i=1}^{n} \alpha_i y_i = 0, \alpha_i \geq 0, \forall i$ with α_i is a Lagrange multiplier. The kernel function $\zeta(x_i, x_j) = x_i \cdot x_j$ is substitued by linear kernel function or polynomial kernel function or Gaussian kernel function so that the weight w is expressed by

$$w = \sum_{i=1}^{n} \alpha_i y_i x_i \tag{16}$$

if linear kernel function, or

$$w = \sum_{i=1}^{n} \alpha_i y_i (a_0 + a_1(x_i \cdot x) + a_2(x_i \cdot x)^2 + \cdots$$
$$+ a_n(x_i \cdot x)^n) \tag{17}$$

if polynomial kernel function, or

$$w = \sum_{i=1}^{n} \alpha_i y_i \left(\exp\left(-\frac{\|x - z\|^2}{2\psi^2} \right) \right) \tag{18}$$

if Gaussian kernel function.

The bias b can be calculated as

$$b = y_k - \sum_{i=1}^{n} \alpha_i y_i \zeta(x_i, x_j), \tag{19}$$

where k is the data index of support vector.

3.3 The Performance Comparison of Kernel Function on Support Vector Machine

This analysis uses the World Disaster Risk Dataset published by the Institute for International Law of Peace and Armed Conflict (IFHV) of the Ruhr-University Bochum in 2022. The World Risk Report is an annual technical report published in German and English on global disaster risk and disaster risk management. The World Risk Report publication includes the World Risk Index, which identifies the risk of extreme natural events that constitute disasters for many countries worldwide. The World Risk Index uses 27 publicly available aggregated indicators to determine the disaster risk of 181 countries worldwide and can be accessed at kaggle.com.

3.4 SVM Classification

SVM classification aims to find the best hyperplane that separates two classes in a feature space with maximum margin. Hyperplane is a decision boundary or separator between different classes. In general, hyperplane is a $(n - 1)$ dimension in a n-dimensional space. For example, if the data is two-dimensional, the hyperplane formed is a line, while if the data is in a three-dimensional space, the hyperplane formed is a plane, and so on. The steps taken in conducting SVM classification analysis include data input, pre-processing data, feature selection, division of training data and testing data, model formation, model training, model validation, and model evaluation.

SVM classification analysis compares linear kernel functions, polynomial kernels, and Gaussian kernels using three data splitting schemes training and testing data 70%-30%, 80%-20%, and 90%-10%. Comparing the performance of different kernel functions aims to evaluate different types of kernels and determine which one gives the best results based on the data set. In the data splitting schemes 70%-30%, 80%-20%, and 90%-10%, there is a possibility that the training and testing data do not represent the proper distribution of the data, which can lead to inaccurate performance assessments. To avoid biased random sample selection on the training data, the k-fold cross-validation method is used.

The k-fold cross-validation method splits the dataset into k equally sized subsets. Each subset is used once as the test set, while the remaining $k - 1$ subsets serve as the training set. This process ensures that every data point is used for testing exactly

Table 1. Comparison of Classification Results Using Linear Kernel Functions, Polynomial Kernels, and Gaussian Kernels

Split Ratio	Kernel	Mean Accuracy	St. Dev. Accuracy	p-value
70:30	linear	0.526477	0.006303	1.247903e-52
	polynomial (d = 4)	0.364656	0.020807	8.696947e-26
	polynomial (d = 5)	0.393747	0.021373	7.169042e-42
	polynomial (d = 6)	0.351254	0.016017	3.827147e-19
	polynomial (d = 7)	0.345287	0.015118	5.619916e-32
	polynomial (d = 8)	0.328880	0.015222	3.297269e-17
	polynomial (d = 9)	0.339322	0.016094	3.609141e-29
	Gaussian	0.522000	0.012058	1.001688e-50
80:20	linear	0.542066	0.007208	3.892815e-64
	polynomial (d = 4)	0.367870	0.028070	8.533074e-30
	polynomial (d = 5)	0.389385	0.032221	5.021827e-39
	polynomial (d = 6)	0.353533	0.017005	2.269181e-24
	polynomial (d = 7)	0.346354	0.018008	1.610738e-38
	polynomial (d = 8)	0.329416	0.007828	4.956321e-25
	polynomial (d = 9)	0.334626	0.010643	2.411910e-33
	Gaussian	0.545999	0.014595	2.260272e-64
90:10	linear	0.540290	0.023087	2.116673e-70
	polynomial (d = 4)	0.342029	0.035783	1.123446e-27
	polynomial (d = 5)	0.381449	0.024223	2.037301e-49
	polynomial (d = 6)	0.356522	0.010207	5.209775e-25
	polynomial (d = 7)	0.351304	0.006711	3.735665e-43
	polynomial (d = 8)	0.328696	0.007906	1.129823e-26
	polynomial (d = 9)	0.339130	0.006610	2.181569e-42
	Gaussian	0.537391	0.024154	3.938239e-74

once and for training $k - 1$ times, leading to a more reliable assessment of the model's performance. The evaluation results from each iteration are averaged to provide a more accurate picture of performance. The selection of the k value is adjusted to the number of data sets. In this SVM classification analysis, $k = 5$ was chosen.

The performance of SVM classification using linear kernel functions, polynomial kernel functions, and Gaussian kernel functions from each scheme 70%-30%, 80%-20%, and 90%-10% is expressed by the average accuracy value, standard deviation of accuracy, and p-value shown in Table 1.

Accuracy is a metric used to measure the extent to which a classifier correctly predicts the class against the total number of samples. The p-value indicates the statistical significance of the classification results. Very small values (< 0.05) indicate statistically significant results.

Based on Table 1, the p-value of all kernel functions show statistically significant results. Still, the linear and Gaussian kernel functions produce better performance than the polynomial kernel functions. Classification using the linear kernel function and the Gaussian kernel function produces an average accuracy of more than 52% and increases with the increase in the training data ratio.

The worst performance is seen from the polynomial kernel results. The accuracy of the polynomial kernel function is much lower than other kernels, ranging from 32% to 39% in each training scheme. There is a tendency for the classification accuracy to not improve as the polynomial degree increases. It can be interpreted that the polynomial kernel function with degrees four to nine is too complex for the data pattern and cannot capture relevant patterns well.

Based on the data split ratios, the results in Table 1 indicate that the accuracy remains relatively consistent across different ratios. This stability may be attributed to the effectiveness of the k-fold cross-validation method. Consequently, the SVM classification applied to the dataset demonstrates that the linear and Gaussian kernel functions outperform the polynomial kernel function in terms of performance. Based on Table 1, it can be concluded that the linear kernel is an excellent choice for this data set because of its high accuracy, good efficiency, and performance stability. The Gaussian kernel provides excellent performance and high efficiency but does not show significant advantages over the linear kernel. The polynomial kernel is unsuitable for this data set because of its low accuracy and high complexity.

Analysis of Support Vector Machine regression, also called Support Vector Regression (SVR), is one of the methods in machine learning used to model the relationship between dependent and independent variables. SVR aims to find a regression function that minimizes the deviation between predictions and actual values while considering the adjusted margin constraints. This section compares regression models using linear kernel functions, polynomial kernels, and Gaussian kernels.

In this SVR analysis, six quantitative variables are used, namely the disaster risk index (world risk index), disaster exposure risk (exposure), vulnerability (vulnerability), sensitivity (susceptibility), coping capabilities (lack of coping capabilities), and the adaptive capabilities of the government and community (lack of adaptive capacities). Based on the understanding of the data characteristics, the variables selected as independent variables are the exposure variable (X_1), the vulnerability variable (X_2), the susceptibility variable (X_3), the lack of adaptive capacities variable (X_4), the lack of coping capabilities variable (X_5), and the world risk index variable (Y) as the dependent variable.

Table 2 shows the parameter estimation results using the linear kernel function. The linear kernel models the linear relationship between the input features of the variables X_1, X_2, X_3, X_4, X_5 and the target output Y. Therefore, SVM regression with a linear kernel explicitly produces coefficients for each feature, showing each feature's direct contribution to the prediction.

Table 2. Parameter Estimation Results Using Linear Kernel Function

Coefficient	Linear Kernel		
	70:30	80:20	90:10
Intercept	15.1910	15.2540	15.2923
X_1	10.5830	10.6364	10.8782
X_2	-0.9017	-0.9030	0.2510
X_3	-1.0020	-1.0200	-1.400
X_4	0.0169	0.0245	0.0609
X_5	-0.0270	-0.0204	-0.1049

The intercept is the average predicted value of the model when all independent variables X_1 to X_5 are zero. The intercept and coefficient values of X_1 to X_5 do not change significantly in the model, even though the data sharing ratio changes.

From the aspect of split ratio, Table 2 shows that the coefficient of X_1 remains stable at all data ratios. In contrast, the coefficients of X_2, X_3, X_4, X_5 show greater fluctuations between split ratios, indicating a weaker relationship or sensitivity to data variation.

Unlike the estimation results using the linear kernel, the estimation results of regression parameters using polynomial kernel functions cannot determine the coefficients of variables X_1, X_2, X_3, X_4, X_5 as shown in Table 3. This is due to the nature of the polynomial kernel function, which projects data into a feature space expanded by a polynomial of a certain degree so that the relationship between features is no longer linear but involves a polynomial combination of the original features.

Table 3 shows that only the intercept parameter is available for the polynomial kernel, with no coefficient values for the independent variables X_1 to X_5. This is a common characteristic of nonlinear kernels, such as polynomial kernels, since the relationship between input and output is not directly represented in explicit linear form.

Table 3. Parameter Estimation Results Using Polynomial Kernel Function

Coefficients	Polynomial Kernel					
	$d = 4$	$d = 5$	$d = 6$	$d = 7$	$d = 8$	$d = 9$
70:30	12.985	13.244	12.873	13.090	12.693	13.004
80:20	13.001	13.263	12.878	13.090	12.819	13.043
90:10	13.041	13.254	12.877	13.085	12.797	13.047

Table 4. Parameter Estimation Results Using Gaussian Kernel Function

Coefficient	Gaussian Kernel		
	70:30	80:20	90:10
intercept	19.675	19.953	20.553

The consistency of the intercept value despite changes in the split ratio indicates that changes in the amount of training data do not significantly affect the intercept estimate. However, since polynomial kernels tend to provide suboptimal performance, the intercept alone is insufficient to indicate the goodness of fit of the polynomial kernel to the data set.

The lack of explicit coefficients for X_1 to X_5 in the polynomial kernel results from how this kernel operates. It transforms the data into a higher-dimensional feature space using polynomial combinations, where the model's predictions are based on the interaction between the kernel function and the support vectors rather than the direct influence of the original input features.

Similar parameter estimates are also produced by SVM Regression using the Gaussian kernel function shown in Table 4. The Gaussian kernel is a distance-based kernel that projects the data into an infinite-dimensional feature space. The absence of coefficients for variables X_1 to X_5 in the Gaussian kernel, as shown in Table 4 is a consequence of the nature of the Gaussian kernel itself, which works in a nonlinear feature space and does not use a linear representation of the original features in the model.

The intercept stability across data ratios indicates that the Gaussian kernel produces a robust model against changes in the amount of data training. The intercept value that does not differ much across various data division ratios indicates that the

Table 5. Comparison of Regression Model Performance Using Linear Kernel Functions, Polynomial Kernels, and Gaussian Kernels

Split Ratio	Kernel	MSE	R^2	MAPE
70:30	linear	4.879	0.957	0.105
	polynomial (d = 4)	45.542	0.481	0.478
	polynomial (d = 5)	40.062	0.542	0.351
	polynomial (d = 6)	49.832	0.432	0.440
	polynomial (d = 7)	44.030	0.509	0.427
	polynomial (d = 8)	47.363	0.470	0.490
	polynomial (d = 9)	48.345	0.457	0.552
	Gaussian	25.102	0.781	0.060
80:20	linear	5.378	0.954	0.089
	polynomial (d = 4)	46.642	0.443	0.403
	polynomial (d = 5)	41.175	0.520	0.310
	polynomial (d = 6)	50.180	0.441	0.388
	polynomial (d = 7)	43.271	0.517	0.401
	polynomial (d = 8)	49.119	0.464	0.501
	polynomial (d = 9)	49.267	0.460	0.503
	Gaussian	31.988	0.729	0.042
90:10	linear	7.049	0.923	0.087
	polynomial (d = 4)	46.564	0.493	0.368
	polynomial (d = 5)	41.072	0.553	0.273
	polynomial (d = 6)	50.065	0.455	0.325
	polynomial (d = 7)	44.175	0.519	0.293
	polynomial (d = 8)	48.361	0.473	0.318
	polynomial (d = 9)	49.018	0.466	0.310
	Gaussian	25.644	0.738	0.034

Gaussian kernel is relatively stable in modelling data patterns even though the size of the training and testing data changes.

Based on Table 2, Table 3, and Table 4, it is concluded that the appropriate kernel function for the data set is the linear kernel function. The recommendation for the proper kernel function is to look at the performance of the linear kernel function, polynomial kernel, and Gaussian kernel based on the Mean Square Error (MSE), coefficient of determination (R^2), and Mean Absolute Percentage Error (MAPE).

Table 5 presents the performance evaluation of the SVM regression model using three kernel types-linear, polynomial, and Gaussian-across different data split ratios (70:30, 80:20, and 90:10). The Mean Squared Error (MSE) represents the average of the squared differences between actual and predicted values; lower MSE values indicate better model performance. The coefficient of determination (R^2) assesses how well the model captures the variability in the target variable, with values closer to 1 signifying stronger explanatory power. Meanwhile, the Mean Absolute Percentage Error (MAPE) reflects the average absolute error percentage relative to the actual values, where a MAPE below 20% denotes a highly accurate model.

Based on Table 5, the linear kernel consistently produces the smallest MSE, the highest R^2, and the smallest MAPE compared to other kernels at all split ratios. The polynomial kernel has the highest MSE, the lowest R^2, and the largest MAPE.

This may be due to the higher complexity of the model so that the model is overfit or not flexible enough to fit the data optimally. The Gaussian kernel performs better than the polynomial kernel but not better than the linear kernel. These results indicate that although the relationship in the data may contain nonlinear elements, the complexity of the Gaussian kernel does not provide a significant advantage over the linear kernel. It can be concluded that the data has a strong linear relationship between the features and the target. Therefore, the linear kernel is more suitable for this data set.

Viewed from the aspect of split ratio, Table 5 shows that the performance decreases with a larger training data ratio. At split ratio 90:10, the performance of all kernels decreases compared to 70:30 or 80:20 ratios. This is evident from the increase in MSE and MAPE and the decrease in R^2. A larger split ratio for the training data reduces the amount of test data, which can cause the evaluation results to be less representative.

Theoretically, the smaller the MSE, the better the model. Based on Figure 1, it appears that the MSE of the linear kernel and the Gaussian kernel are close to 0 for each scheme, while for

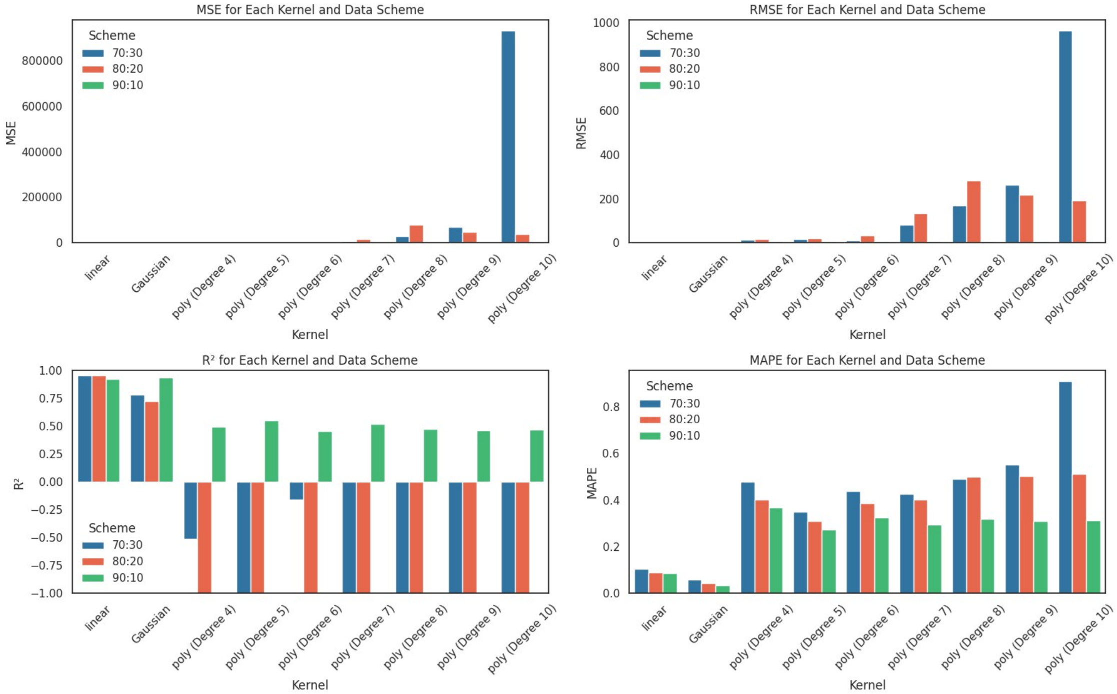

Figure 1. Evaluation of SVR Model Using Linear Kernel Function, Polynomial Kernel, and Gaussian Kernel

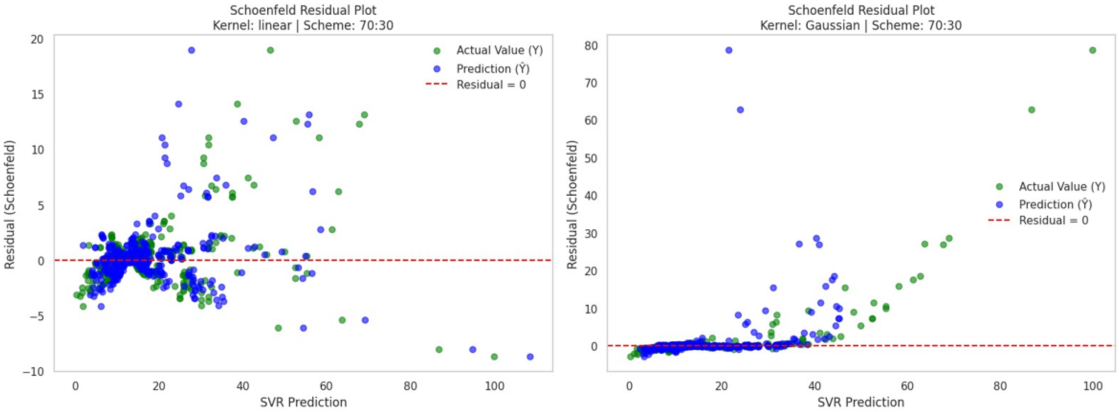

Figure 2. Schoenfeld Plot of Linear Kernel Function and Schoenfeld Plot of Gaussian Kernel Function

the polynomial kernel, the MSE value increases as the degree of the polynomial increases. The polynomial kernel of degree 9 shows a very large MSE, especially in the 70:30 scheme, which means the model is overfitting or fails to capture the pattern. Based on the MSE value, the linear kernel function is the best choice because it produces the lowest mean squared error, indicating a strong linear relationship in the data. Polynomial kernels with high degrees are unstable and less suitable for this dataset.

A low value (closer to 0) is better for the MAPE metric. Based on Figure 1, the linear and Gaussian kernels have low and stable MAPE values for each scheme, while the polynomial kernel tends to show a high MAPE, especially at degree 9. Thus, relatively (in terms of percentage error), the linear and RBF kernels show excellent and consistent performance across data partitioning schemes.

When analyzed based on the R^2 value, a value close to 1 indicates a good fit. The linear kernel function for each scheme provides R^2 close to 1, making the model very good. The Gaussian kernel function also performs well but is more sensitive to the proportion of data. The polynomial kernel (degree 4–9) produces a negative R^2, meaning its performance is worse than the mean-only model. Based on Figure 1, only the linear and Gaussian kernels are worthy of consideration for this model. In contrast, the polynomial kernel is unsuitable and causes underfitting or overfitting.

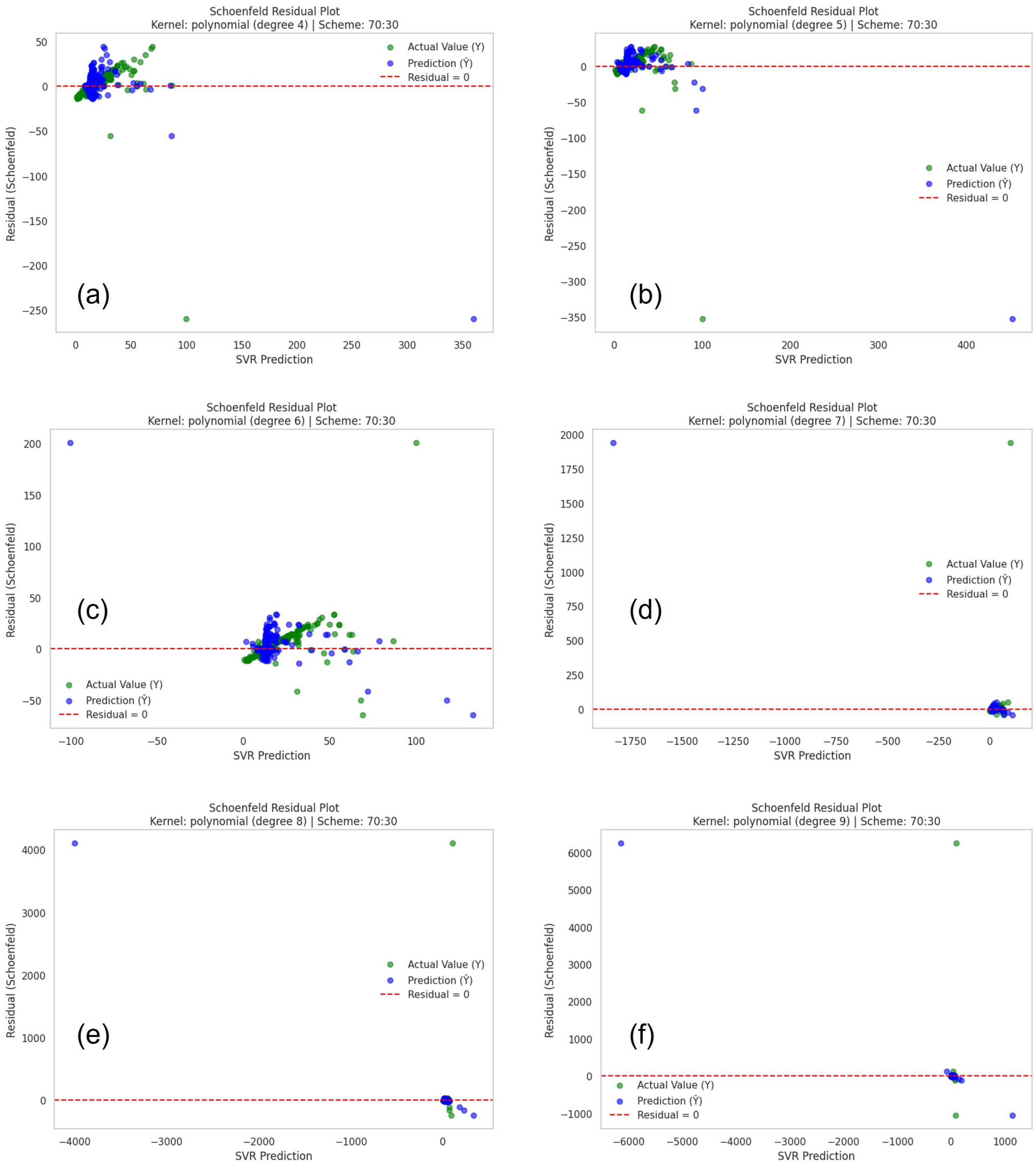

Figure 3. Schoenfeld Plot of Polynomial $d = 4$ (a), Schoenfeld Plot of Polynomial $d = 5$ (b), Schoenfeld Plot of Polynomial $d = 6$ (c), Schoenfeld Plot of Polynomial $d = 7$ (d), Schoenfeld Plot of Polynomial $d = 8$ (e), Schoenfeld Plot of Polynomial $d = 9$ (f)

Based on Figure 1, the best kernel function is linear, as it produces the lowest MSE, RMSE, and MAPE and the highest and most stable R^2. The Gaussian kernel function performs reasonably well but is somewhat inconsistent across schemes. High-degree polynomial kernel functions are not recommended as they perform poorly across all evaluation metrics. The 90:10 scheme tends to produce the best results due to the larger training data, but the linear kernel still holds the best performance consistency.

The residual analysis of each kernel function is displayed graphically using the Schoenfeld plot. Based on Figure 2, the green dots are the actual data values, while the blue dots are the predicted values. Based on Figure 2, it appears that most of the residuals are around zero, especially for small SVM Regression predictions (< 30). The residuals become very large, especially for higher SVM Regression predictions (> 40). Based on Figure 3, it can be seen that as the degree of the polynomial increases (from 4 to 9), the residual values become more extreme, both in positive and negative scales. At a larger degree (for example, 8 or 9), the SVM Regression prediction has a value that is very far from the actual data, which causes the residual to jump drastically. This shows that the model is becoming more complex and can cause overfitting, so the SVM Regression prediction becomes less good. So, numerically and graphically, the linear kernel function is more appropriate to be implemented on the data set used.

4. CONCLUSIONS

Based on the research and discussion results, it is concluded that the construction of Hilbert probability space with unit elements is not unique. In implementing Reproducing Kernel Hilbert Space, the same kernel function can produce different feature mapping dimensions. The analysis reveals that a finite linear combination of kernel functions retains the kernel property, which supports the construction of both polynomial and Gaussian kernel functions. In applying the dataset, the linear kernel function emerges as the most suitable option, consistently delivering the most accurate and stable results across all data split ratios. This suggests the presence of a strong linear relationship within the dataset. The Gaussian kernel offers the flexibility to handle non-linear elements but is not better than the linear kernel. The polynomial kernel performs poorly because its complexity does not match the data pattern. This study provides strong evidence for the suitability of linear kernels in modeling disaster risk panel data. However, questions remain regarding the behavior of kernels in time-evolving models and whether hybrid kernels could improve long-term generalization. Future work is needed to address these challenges.

5. ACKNOWLEDGEMENT

The author wants to thank the Universitas Lampung, supervisor, co-supervisors, and reviewers for their contributions to producing a good article. This research was funded by Directorate of Research, Technology, and Community Service, Directorate General of Higher Education, Research, and Technology, Ministry of Education, Culture, Research, and Technology in accordance with Research Contract Number: 076/C3/DT.05.00/PL/2025.

REFERENCES

Akgül, A. (2019). Reproducing Kernel Hilbert Space Method Based on Reproducing Kernel Functions for Investigating Boundary Layer Flow of a Powell–Eyring Non-Newtonian Fluid. *Journal of Taibah University for Science*, **13**(1); 858–863

Anjani, S. D., W. Widiarti, B. H. S. Utami, M. Usman, and V. A. Handayani (2024). Georaphically Weighted Ridge Regression Modelling on 2023 Poverty Indicators Data in the Provinces of West Kalimantan and Central Kalimantan. *Integra: Journal of Integrated Mathematics and Computer Science*, **1**(3); 73–80

Arqub, O. A., M. Al-Smadi, and N. Shawagfeh (2013). Solving Fredholm IntegroDifferential Equations Using Reproducing Kernel Hilbert Space Method. *Applied Mathematics and Computation*, **219**(17); 8938–8948

Awan, D. A., L. G. C. Renato, M. Yukawa, and S. Stanczak (2020). Adaptive Learning for Symbol Detection: A Reproducing Kernel Hilbert Space Approach. In *Machine Learning for Future Wireless Communications*. pages 197–211

Azarnavid, B., F. Parvaneh, and S. Abbasbandy (2015). Picard-Reproducing Kernel Hilbert Space Method for Solving Generalized Singular Nonlinear Lane-Emden Type Equations. *Mathematical Modelling and Analysis*, **20**(6); 754–767

Baltagi, B. H., P. H. Egger, and M. Kesina (2012). Small Sample Properties and Pretest Estimation of a Spatial Hausman-Taylor Model. In *Advances in Econometrics*, volume 29. pages 215–236

Berlinet, A. and C. Thomas-Agnan (2004). *Reproducing Kernel Hilbert Spaces in Probability and Statistics*. Springer, New York, USA

Bowman, A. W. and A. Azzalini (1997). *Applied Smoothing Techniques for Data Analysis: The Kernel Approach*. Clarendon Press, Oxford, USA

Cervantes, J., G.-L. Farid, L. Rodríguez-Mazahua, and A. Lopez (2020). A Comprehensive Survey on Support Vector Machine Classification: Applications, Challenges, and Trends. *Neurocomputing*, **408**(1); 189–215

Chakraborty, S. (2012). Bayesian Multiple Response Kernel Regression Model for High Dimensional Data and Its Practical Applications in Near Infrared Spectroscopy. *Computational Statistics and Data Analysis*, **56**(9); 2742–2755

Chen, S. B., S. Soradi-Zeid, H. Dutta, M. Mesrizadeh, H. Jahanshahi, and Y. M. Chu (2021). Reproducing Kernel Hilbert Space Method for Nonlinear Second Order Singularly Perturbed Boundary Value Problems with Time-Delay. *Chaos, Solitons and Fractals*, **144**(1); 1–10

Ghojogh, B., A. Ghodsi, F. Karray, and M. Crowley (2021). Reproducing Kernel Hilbert Space, Mercer's Theorem, Eigenfunctions, Nystrom Method, and Use of Kernels in Machine Learning: Tutorial and Survey. *CoRR*, **2106**; 1–31

Ghosh, S., A. Dasgupta, and A. Swetapadma (2019). A Study on Support Vector Machine Based Linear and Non-Linear Pattern Classification. In *Proceedings of the International Conference on Intelligent Sustainable Systems*, volume 1. pages 24–28

Lestari, W., W. Widiarti, B. H. S. Utami, M. Usman, and V. A. Handayani (2024). Robust Panel Data Regression Analysis Using the Least Trimmed Squares (LTS) Estimator on Poverty Line Data in Lampung Province. *Integra: Journal of Integrated Mathematics and Computer Science*, **1**(2); 35–42

Mohan, L., J. Pant, P. Suyal, and A. Kumar (2020). Support Vector Machine Accuracy Improvement with Classification. In *Proceedings 12th International Conference on Computational Intelligence and Communication Networks*, volume 12. pages 477–481

Muthukrishnan, S., H. Krishnaswamy, S. Thanikodi, D. Sundaresan, and V. Venkatraman (2020). Support Vector Machine for Modelling and Simulation of Heat Exchangers. *Thermal Science*, **24**(1B); 499–503

Paiva, A. R. C., I. Park, and J. C. Principe (2019). A Reproducing Kernel Hilbert Space Framework for Spike Train Signal Processing. *Neural Computation*, **21**(2); 424–449

Paulsen, V. I. and M. Raghupathi (2009). *An Introducing to The Theory of Reproducing Kernel Hilbert Spaces*. University of Houston, Texas, USA

Pisner, D. A. and D. M. Schnyer (2020). Support Vector Machine. In *Machine Learning: Methods and Applications to Brain Disorders*. Elsevier Inc., New York, USA

Rampisela, T. V. and Z. Rustam (2018). Classification of Schizophrenia Data Using Support Vector Machine (SVM). *Journal of Physics: Conference Series*, **1108**; 1–7

Rincón, M. and M. D. Ruiz-Medina (2012). Wavelet-RKHS-Based Functional Statistical Classification. *Adv Data Anal Classif*, **6**; 201–217

Rosipal, R. and L. J. Trejo (2001). Kernel Partial Least Square Regression in Reproducing Kernel Hilbert Space. *Journal of Machine Learning Research*, **2**; 97–123

Shawe-Taylor, J. and N. Cristianini (2004). *Kernel Methods for Pattern Analysis*. Cambridge University Press, New York, USA

Shukla, S., N. Badal, and B. K. Thakur (2023). *Sustainable Computing: Transforming Industry 4.0 to Society 5.0*. Springer, New York, USA

Small, C. G. and D. L. McLeish (1994). *Hilbert Space Methods in Probability and Statistical Inference*. John Wiley and Sons, New Jersey, USA

Tuo, R., S. He, A. Pourhabib, Y. Ding, and J. Z. Huang (2023). A Reproducing Kernel Hilbert Space Approach to Functional Calibration of Computer Models. *Journal of the American Statistical Association*, **118**(542); 883–897

Utami, B. H. S., M. Usman, and F. Warsono (2023). The Form of σ-Algebra on Probability Hilbert Space. *Mathematics and Statistics*, **11**(3); 446–453

Yang, H. (2016). *Learning Methods in Reproducing Kernel Hilbert Space Based on High-Dimensional Features*. Ph.D. thesis, University of California, California, USA

Zhu, W., Y. Song, and Y. Xiao (2020). Support Vector Machine Classifier with Huberized Pinball Loss. *Engineering Applications of Artificial Intelligence*, **91**; 1–9