分类号 密 级

U D C 学校代码 10500



****

**硕士学位论文**

Computer Sciences - Master

题 目： 针对有异常值的 Naive Bayes 分类的优化鲁棒核密度估计

###### 英文题目：Optimised Robust Kernel Density Estimation for Naive Bayes Classification with Outliers

**学位申请人姓名：BOLI BI IRITIE A-D**

**硕士学位课程：Computer Sciences**

**指导教师姓名：Dr. Chenghao Wei**

二○二 年五月

分类号 密 级

U D C 学校代码 10500



****

**硕士学位论文**

题 目:针对有异常值的 Naive Bayes 分类的优化鲁棒核密度估计

英文题目: Optimized Robust Kernel Density Estimation for Naive Bayes Classification with Outliers

研究生姓名（签名）: BOLI BI IRITIE ARNAUD-DONATIEN

指导教师姓名（签名）: Chenghao Wei 职 称: Doctor

申请学位学科名称 Computer Science 学科代码: 211cS1

论文答辩日期 学位授予日期

学院负责人（签名）:Computer School

评阅人姓名 评阅人姓名

年 月 日



学位论文原创性声明和使用授权说明

**Originality Statement**

I solemnly declare that the dissertation submitted is the result of my research work conducted independently under the guidance of my supervisor. Except for what has been marked and cited in the text, this dissertation does not contain any research results that have been published or written by any other individual or collective. Individuals and collectives who have contributed to the research of this paper have been clearly indicated in the text. The legal consequences of this statement will be borne by me.

Signature：                               Date：    　 年  　 月 　  日

**Authorization for the use of copyright in the dissertation**

The author of this dissertation fully understands the regulations of the university concerning the retention and use of the dissertation, that is, the university has the right to retain and send copies of the dissertation and its electronic version to the relevant departments or institutions of the state, and to allow the dissertation to be inspected and borrowed. I hereby authorise Hubei University of Technology to compile all or part of this dissertation into relevant databases for searching, and to preserve and compile this dissertation by means of reproduction such as photocopying, microfilming or scanning.

Signature of the author：          　　　　     Signature of instructor：

Date：     年    月    日         　　　  日期：     年    月     日

# 摘 要

**关键词：**

# Abstract

In the field of machine learning, one of the most significant challenges consists in the difficult endeavour of correctly categorising data when there are outliers present. This research study presents the concept of "Optimised Robust Kernel Density Estimation" (ORKDE), which is a one-of-a-kind method with the objective of enhancing the effectiveness of the Naive Bayes classifier under circumstances that are marked by the presence of outliers. In order to determine the bandwidth that is most suited for a given application, the ORKDE framework makes use of powerful M-estimation techniques. These methods include the Iterative Reweighted Least Squares approach and the Harris Hawks Optimisation. A comprehensive Cross-Validation Curve that takes into account Unbiased Cross-Validation, Biassed Cross-Validation, and Bootstrap techniques is required in order to implement the strategy that is currently being proposed.

The ORKDE framework has been put through extensive design and testing processes to ensure its quality. The empirical data suggests that the method is robust in the presence of outliers and has the potential to significantly improve the accuracy of classification. Additionally, the method has the ability to significantly reduce the number of false positives. The ORKDE method improves the effectiveness of the classic Naive Bayes classifier in the face of challenging datasets that include noise and complexity. This is accomplished by optimising both the kernel density estimate and the bandwidth selection processes.

The current investigation presents a comprehensive analysis of the ORKDE method, which includes a painstaking investigation of the technique's fundamental components and the connection between them. The extensive empirical evidence illustrates the use of this technique across a variety of datasets, as well as its superior effectiveness when compared to more conventional methods.

**Keywords**: Robust Kernel Density Estimation (RKDE), Naïve Bayes Classifier, Outliers,Harris Hawk Optimization (HHO),Bandwidth Selection, Iterative Reweighted Least Squares, Robust M-estimation, Cross-Validation Curve, Machine Learning

# Table Contents

摘 要 I

Abstract II

Table Contents III

1. Introduction 1

1.1. Background and motivation 1

1.2. Research Objectives 2

1.3. Thesis Structure 2

2. Literature Review 4

2.1. Machine Learning and Intelligence Data Analysis 4

2.2. Kernel Density Estimation (KDE) 6

2.2.1. Gaussian KDE and its Applications 6

2.2.2. Limitations of Traditional KDE 8

2.3. Robust Kernel Density Estimation (RKDE) 8

2.3.1. M-estimation and Hampel Function 9

2.3.2. Iterative Reweighted Least Squares (IRLS) 11

2.4. Naïve Bayes Classifier 12

2.4.1. Baisc Naïve Bayes Classifier 12

2.4.2. Challenges with Noisy and Tough Datasets 13

3. Methodology 14

3.1. Robust RKDE with IRLS 14

3.2. Harris Hawk Optimization (HHO) for bandwidth selection 15

3.2.1. Harris Hawk Optimization 15

3.2.2. Description HHO 16

3.2.2.1. The Process of HHO 20

3.2.3. Selecting Bandwidth with HHO 21

3.3. Complete Cross-Validation Curve (CCV) 22

3.3.1. Unbiased Cross-Validation 22

3.3.2. Biased Cross-Validation (BCV) 23

3.3.3. Bootstrap 24

3.3.4. Importance of using CCV in RKDE and Intelligence Data Analysis 25

3.4. Integration of Robust KDE and Bandwith selection into Naïve Bayes classifier/ Enhancing Naïve Bayes Classifier with Robust KDE and Bandwidth Optimization / Robust Kernel Density Estimation and Bandwidth Selection in Naïve Bayes Classification 26

4. Experimental Setup 28

4.1. Dataset Description 28

4.1.1. Synthetic data 29

4.1.2. Real-World data 30

4.2. Implementation Details 30

4.3. Performance metrics 31

4.3.1. Accuracy 31

4.3.2. Precision 32

4.3.3. Recall 32

4.3.4. F1-Score 32

4.3.5. Area Under the Curve (AUC) 32

5. Results and Discussion 33

5.1. Evaluation of Optimized Robust KDE 33

5.2. Comparison with traditional Naïve Bayes Classifier 36

6. Conclusion and Future Work 39

6.1. Summary of findings 39

6.2. Contributions to the field 39

6.3. Limitation and Future direction 40

7. References 42

Appendices 44

A. List of papers published by the author during his degree study 44

B. Python code display 44

# Introduction

# Background and motivation

In recent years, there have been major improvements in the field of machine learning, notably in the domain of intelligence data analysis. These advancements have been driven by the rise of artificial intelligence. As a result of their capacity to analyse big datasets and make effective use of robust computational resources, machine learning algorithms have seen widespread use across a wide variety of industries, including healthcare and finance, amongst others. These algorithms have proven to be effective in solving complex problems that were previously thought to be intractable. Our capabilities in the areas of decision-making, automation, and predictive analytics have been radically revolutionised as a direct result of the broad adoption of machine learning. In spite of this, the growing reliance on machine learning techniques has resulted in a heightened awareness of the vulnerabilities associated with these methods, particularly in the face of noisy and complicated datasets. The data collected from the real world frequently have a variety of flaws, such as outliers, problems in measurement, and occasions where statistics are missing. These defects, if they are not addressed, can have a significant effect on the dependability and effectiveness of machine learning models, which can lead to results that are not as good as they could be.

The ability to accurately analyse and simulate the distributions of data is a crucial prerequisite for a wide variety of machine learning approaches. Kernel Density Estimation, also known as KDE, is an important technique in this area because it gives researchers the capacity to depict the data distributions that lie beneath the surface. A broad variety of different applications require anomaly detection, classification, and clustering, all of which are handled by the KDE technique. This method plays an important part in all of these processes. The efficiency of machine learning models is highly dependent on the accuracy of density estimates, which serve as the foundation for fundamental processes like as classification, decision-making, and risk assessment. Conventional KDE methods, on the other hand, usually run into problems when confronted with datasets that contain outliers and noisy observations. The method of estimating density is susceptible to being skewed by anomalous data points, which are also called outliers on occasion. This is because of the fact that these data points have the ability to have a disproportionate influence. As a direct consequence of this, it is possible that skewed models will be produced, as well as poor results in the future tasks. When it comes to strengthening the robustness and reliability of machine learning models, the challenge of tackling this difficulty and refining the Kernel Density Estimation technique bears major significance.

Data points that are considered to be outliers are those that deviate significantly from the norm of the other data points. Errors in measurement, manipulation of data, or the occurrence of extraordinary occurrences could all be potential causes of these problems. Outliers have important informative value in a variety of real-world contexts and should not be ignored; rather, they should be effectively included into the process of modelling. However, it is essential to realise that these methods have the potential to change the underlying distribution of the data. This can lead to erroneous estimations of density and, as a result, faulty predictions from the model. It is for this reason that it is vital to acknowledge this potential risk.

# Research Objectives

The primary objective of this research project is to make a significant contribution to the development of accurate Kernel Density Estimation and to the application of this methodology to the analysis of intelligence data. The following are some of the specific goals that our research tries to achieve:

To make a certain process or system more efficient or effective, or both, by increasing their overall potential. The purpose of this research is to create and improve methods of resilient Kernel Density Estimation that are capable of accurately estimating data densities in situations when there are both outliers and noisy data points present. The term "robust" refers to methods that are less susceptible to being affected by the influence of outliers and provide density estimates that are more reliable.

Maximising the Efficiency of Bandwidth Selection: to optimise the selection of bandwidth parameters, which play an essential part in Kernel Density Estimation, it is proposed to make use of the Harris Hawk Optimisation method. This would allow for maximum efficiency in the process. The purpose of this optimisation is to improve the precision and adaptability of KDE when it is applied to a variety of datasets. Within the framework of KDE, the selection of bandwidth is a critical and significant task, and its optimisation has the ability to improve the accuracy and precision of density estimates.

The purpose of this research is to investigate and put into practise different strategies that can reduce the impact that outliers have on classification prediction. In particular, the focus will be on studying several strategies that can be used to reduce the negative effects that are caused by outliers in KDE-based models, with a special emphasis being placed on the Naive Bayes classifier. In the field of machine learning, classification tasks are frequently seen, and the ability to give solid predictions even when outliers are present is of significant practical value.

# Thesis Structure

This thesis is structured as follows:

**Chapter 2**: The Research Review provides an in-depth analysis of the most recent findings in the fields of machine learning, Kernel Density Estimation, robust statistics, and the challenges that are posed by datasets that contain noise. In order to determine the current state of understanding in these areas, we make use of significant previous research and current scholarly projects.

**Chapter 3:** The methodology section includes a full explanation of the strategies that were utilised in this study. These methodologies include the enhancement of robust KDE procedures, the utilisation of the Harris Hawk Optimisation algorithm, and the merging of these approaches into the Naive Bayes classifier. Within the scope of this thesis, we will discuss the essential technical ideas that form the basis of our research.

**Chapter 4**: In the section under "Experimental Setup," a thorough summary of the datasets that were used, the technical implementation details, and the performance criteria that were used to evaluate the efficacy of the proposed methodologies is provided. In order to determine whether or not our approaches are effective, conducting thorough experiments is of the utmost importance and should be implemented as soon as possible.

**Chapter 5:** The section of the report headed "Results and Discussion" provides a concise summary of the outcomes of the tests that were carried out, an analysis of the repercussions of those outcomes, and an evaluation of the influence that bandwidth selection has on the performance of categorization. We do a comprehensive examination of the practical repercussions of our study by utilising both quantitative analysis and qualitative discussions.

**Chapter 6:** In conclusion, the findings of this investigation have supplied important new perspectives on the matter at hand. Moving forward, there are a number of different directions that future research could take, each of which could contribute to an even deeper comprehension of the topic at hand. The current paper gives a complete summary of the research's contributions, delineates its inherent restrictions, and proposes prospective routes for additional inquiry within the domain of robust KDE for machine learning. The focus of the current discussion will be on the implications of our study as well as the potential it possesses to advance the field of intelligence data analysis.

**Chapter 7:** The section on references has a comprehensive collection of sources, which includes significant publications that have served as a source of direction for the current investigation. We would want to convey our appreciation for the significant contributions that members of the broader scientific community have made in shaping and enhancing our understanding of the topic at hand.

This thesis intends to provide a contribution to the advancement of robust KDE techniques, with the ultimate goal of boosting the capabilities of machine learning models in effectively managing difficult and noisy datasets. This contribution will be made in the form of a thesis. A more in-depth understanding of intelligent data analysis within the context of machine learning will be made possible by the coming chapters, which will investigate the intricacies of the aforementioned methods and the empirical evidence supporting them.

# Literature Review

The literature review that is included in this thesis provides a detailed study of the fundamental principles as well as recent achievements in the fields of intelligent data analysis and machine learning. This chapter lays the groundwork by conducting an analysis of the underlying principles and challenges associated with these domains. In doing so, it lays the groundwork for an in-depth research of KDE and the resilient variations that it offers.

# Machine Learning and Intelligence Data Analysis

The field of study known as machine learning, which is classified as an area of artificial intelligence (AI), has seen a period of tremendous development in recent decades, leading to an extension of its scope. Without the need for explicit programming, computers are able to acquire information from data, recognise patterns, and make informed judgements through the process of machine learning, which involves a varied range of approaches. The development of machine learning can be summed up in a few key points as follows:

**Early Developments**: Historically speaking, the beginnings of machine learning can be traced back to the middle of the 20th century. During this time period, significant pioneers such as Alan Turing and Arthur Samuel created the essential ideas of rule-based learning and game-playing algorithms, which laid the groundwork for the development of machine learning. These preliminary breakthroughs served as a solid groundwork for the later inquiries.

**Statistical Learning**: The area of machine learning has been through a gradual change, changing into a statistical discipline that extensively includes principles from probability theory and statistical procedures. This transformation has occurred as a result of the field's progression towards becoming a statistical discipline. In the field of pattern recognition and prediction, techniques such as linear regression and decision trees have amassed a large amount of prominence in recent years.

**The Era of Neural Networks**: Within the subject of machine learning that is known as "deep learning," artificial neural networks became a major paradigm during the 1980s and 1990s. However, the potential of their powers was hampered by the limits imposed by poor computer resources and an insufficient amount of data. This prevented them from reaching their full potential.

**The Big Data Revolution**: The discipline of machine learning has seen a huge revolution in the 21st century, primarily as a result of the widespread availability of extensive datasets and the notable advancements in processing power. As a consequence of this, the revival of neural networks in the form of deep learning has emerged as a dominant force in a variety of fields, including image recognition, natural language processing, and autonomous vehicle technology, to name just a few.

**Interdisciplinary Impact:** Machine learning has expanded beyond its traditional limitations and is now being utilised in a variety of other fields, including healthcare (for the purposes of diagnosis and the development of medications), finance (including algorithmic trading and the detection of fraud), and natural language processing (including chatbots and language translation). [3,4]

The study of intelligence data as it relates to the discipline of machine learning is of great significance in the extraction of actionable insights, the facilitation of informed decision-making, and the automation of sophisticated procedures in a variety of different industries. This is due to the fact that intelligence data may be used to learn from past mistakes. A few of the essential factors that contribute to its relevance are as follows:

**Decision Support:** The examination of data gathered by intelligence services provides decision-makers with valuable insights that are gleaned from huge databases. Insights that are driven by data play a critical part in enabling businesses in a variety of industries, such as healthcare, finance, and marketing, to make educated decisions that improve overall performance and operational efficiency.

**Predictive Modelling:** The approaches of machine learning make it possible to create prognostic models, which can predict future occurrences or patterns. This holds important significance in a variety of contexts, including the forecasting of meteorological conditions, the prediction of market trends, and the monitoring of disease outbreaks.

**Anomaly Detection:** The detection of anomalies or outliers in data is an important task that has applications in a variety of areas, including the prevention and detection of fraud, the management of network security, and the quality control of products. The algorithms that are used in machine learning are quite good at recognising patterns that deviate from the established norm.

**Pattern Recognition:** The models that are used in machine learning are quite good at recognising complex patterns that are included inside datasets. This skill is used in the fields of picture identification, speech recognition, and natural language comprehension, leading to breakthroughs like as driverless vehicles and virtual personal assistants.

**Personalization:** The exploitation of intelligence data analysis is an essential component of the driving force behind recommendation systems, which are intended to provide individualised content and product options. These technologies have the potential to enhance user experiences across a variety of platforms and industries, including e-commerce, streaming services, and online advertising, among others.[5]

The increasing prevalence of data-driven decision-making across a variety of business sectors highlights the ever-increasing need of dependable and precise data analysis tools. Because the method of Kernel Density Estimation has such substantial weight when it comes to the modelling of data distributions, we have decided to make it the primary focus of our investigation for this thesis.

In the following sections of this chapter, we are going to take a more in-depth look at the concepts and problems that are associated with KDE, notably in the context of robust data analysis.

# Kernel Density Estimation (KDE)

Estimating the probability density function (PDF) of a continuous random variable can be accomplished with the use of a commonly used statistical method known as the KDE procedure. This method plays an important role in a variety of applications, including smoothing the data, estimating the density of the data, and locating outliers. In this section, we will examine the different applications of the Gaussian KDE, which is often regarded as one of the variations that is used the most frequently. In addition to this, we are going to investigate the limitations that are connected with traditional KDE approaches. [1]

# Gaussian KDE and its Applications

The goal of the non-parametric method known as the Gaussian kernel density estimation (KDE) is to estimate the probability density function (PDF) of a certain dataset. Applying a Gaussian (normal) distribution, also known as a kernel, to each individual data point and then combining those results to generate an overall density estimate is what this process entails. The Gaussian kernel function, denoted as , is expressed as:

Where:

* is the kernel function.
* represents the distance from the data point of interest.

The Gaussian KDE estimate for a given data point with a dataset is calculated as follows:

Where:

* stands for the KDE estimate of the PDF at point .
* is the total number of data points in the dataset.
* signifies individual data points within dataset .
* is the bandwidth parameter, responsible for controlling the level of smoothing and is a pivotal factor in KDE as it determines the width of the kernels.

Gaussian KDE has gained widespread application in various domains:

**Probability Density Estimation:** Because it can provide a non-parametric estimation of the PDF, the Gaussian KDE is a highly significant tool in the field of statistical analysis and hypothesis testing. This is owing to the fact that it can estimate the PDF.

**Data Smoothing:** The process described above is utilised in order to lessen the effect that noise has on the data. As a direct consequence of this, high-frequency noise is diminished, while basic patterns in temporal data are brought to the fore.

**Data Visualization:** The Gaussian KDE method is an essential component in the process of creating visually appealing density plots, which are also referred to as kernel density plots. The objective of these plots is to provide a visual representation of the data's distribution, which is why they are used so frequently.

**Outlier Detection:** The Gaussian KDE technique helps in the discovery of outliers in datasets by developing a model of the underlying distribution of the data and detecting data points that display major deviations from this model. This allows the technique to assist in the identification of data points that are significantly different from the model.[7,9]

# Limitations of Traditional KDE

While Gaussian KDE is a powerful tool, it is not without its limitations:

**Bandwidth Selection:** The output of the KDE is significantly impacted by the bandwidth setting *(h*) that is used. An estimate with an excessively narrow bandwidth will have a higher tendency to be overfit, while an estimate with an excessively broad bandwidth will have a higher tendency to be overly smoothed. The selection of an appropriate bandwidth can be a difficult and time-consuming task.

**Curse of Dimensionality:** The more dimensions there are in the data, the less successful KDE becomes at analysing it. As the number of dimensions rises, there is a proportional increase in the rate at which the volume of the kernel decreases. This causes sparsity to become more apparent in spaces that have a high number of dimensions.

**Computational Demands**: It is possible that the computing requirements of KDE, particularly in high-dimensional domains, may be enormous; as a result, its applicability to large datasets will be limited.

**Edge Effects:** The estimate of KDE towards the boundaries of the data domain may display bias due to the absence of data points beyond the boundary, which is essential for the development of valid kernels. This is because having data points beyond the boundary is necessary for the formation of valid kernels.

**Sensitivity to Data Density:** The density of the data as well as its distribution can have a significant impact on KDE's performance, which can vary significantly as a result. A low performance might be the result of using sparse or multimodal datasets.

**Boundary Issues:** Due to the fact that there are boundaries or discontinuities in the data, it is possible for the estimates to be erroneous because of the possibility of undersmoothing or oversmoothing.

In light of the limitations outlined above, researchers have conducted researches into robust modifications of the KDE technique and have created ways to optimise the choice of bandwidth. In the next chapters of this thesis, we will investigate several methods for accurate kernel density estimation as well as innovative approaches to overcome these restrictions. [2]

# Robust Kernel Density Estimation (RKDE)

Estimating probability density functions based on observed data may be done in a non-parametric manner with the help of the KDE technique. The traditional KDE approach has a high propensity to be affected by outliers and may produce density estimates that are skewed when there is an abundance of values that are at the extreme end of the scale. The use of RKDE has emerged as a useful method in order to address this issue. This technology offers resistance against the outliers and noise that are present in the data [10]. It is possible to formulate an expression for it using the following formula:

Let be a dataset with n data points, and represents the kernel function. The RKDE estimate of the probability density function at point is given by:

where:

* is the bandwidth parameter, determining the width of the kernel window.
* signifies individual data points within dataset
* is the weight assigned to each data point based on robust M-estimation, such as the Hampel function.

# M-estimation and Hampel Function

M-estimation is a statistical method that is widely utilised in the field of robust statistics for the aim of estimating the parameters of a particular model. This is accomplished through the utilisation of robust statistics. The major goal of this research is to determine the ideal values for the study's parameters that will maximise a predetermined objective function that is more generally referred to as the "M-estimator." The presence of outliers is one of the key challenges that researchers in the field of resilient density estimation face. Outliers have the potential to have a significant effect on the level of precision achieved by traditional KDE methods [11].

In order to overcome this specific challenge, we suggest using the Hampel function, which is a reliable weight function that is utilised in M-estimation procedures. In doing so, we will be able to address the issue at hand. On the other hand, the M-estimation approach provides us with a large number of objective functions, such as *Huber, tuTukey's biweight and bisquare* functions. In robust M-estimation, the Huber function is a frequently used choice since it allows for the downweighting of outliers while still maintaining the effect of inliers. The definition of the formula is as follows:

where δ represents the tuning parameter, the Huber function strikes a balance between robustness and sensitivity, making it suitable for mitigating the impact of outliers [15].

The Huber function is expanded upon by the Hampel function, which is another reliable M-estimator, by the addition of a parameterized threshold. This threshold modifies the estimator's behaviour in reaction to outliers and its capacity to diminish or "trim" extreme values, thereby reducing the impact that outliers have on the estimate procedure. Hampel is an extension of the Huber function [13]. As was already stated,

where x represents the residual (difference between the data point and the estimated density), and δ is a tuning parameter that determines the robustness of the estimator.

For |x| ≤ δ, the function behaves quadratically, penalizing small residuals.

For |x| > δ, the linear behaviour of the function provides a more robust weight To bigger residuals, hence lowering the influence that outliers have on the estimate of the density [17]. Despite the fact that the biweight and bisquare functions of the Tukey package, in addition to the Cauchy function, provide other methods for obtaining robustness, the Huber and Hampel packages continue to be preferred alternatives. These functions augment the M-estimation process with varying degrees of non-linearity, providing a variety of strategies for reducing the importance of outliers and increasing the robustness of the estimator. In conclusion, the selection of the M- estimation function has a significant impact on the performance as well as the robustness of KDE techniques in the treatment of outliers. Because each function provides a distinct trade-off between robustness and sensitivity, the decision should be made in line with the characteristics of the data and the degree of outlier attenuation that is required.

The RKDE is characterised as the following components: the integration of the Hampel function within the KDE framework:

where:

* is the bandwidth parameter, determining the width of the kernel window.
* signifies individual data points within dataset
* is the weight assigned to each data point based on robust M-estimation, such as the Hampel function.
* is the Hampel function, which assigns weights to data points based on their distance from .

The width of the kernel is controlled to a large extent by the bandwidth parameter *h*, which also has an important bearing on the degree to which the density estimate is smooth. It requires careful selection in order to achieve an equilibrium between bias and variation in the estimate.

The RKDE takes into consideration the distance each data point is from the evaluation point x when calculating the weighted contributions of each point in the dataset. The use of a kernel function, which helps to even out the rough edges of the contributions, is what makes this possible. In contrast to more traditional KDE methods, RKDE is able to provide estimations of density that are more accurate even when presented with data distributions that depart from the norm or with outliers in the data. This is what makes RKDE so resilient [10,12,13,14].

It is essential to keep in mind that various RKDE implementations can use different kernel functions and robustness approaches, which would lead to modifications in the approach's particular formulation.

# Iterative Reweighted Least Squares (IRLS)

In the subject of robust statistics, specifically in the context of robust density estimation, the IRLS technique is a frequently used example of a numerical optimisation methodology that is put to use. During this iterative procedure, parameter estimations are refined by being brought up to date in accordance with the weighted least squares criteria. M-estimation problems, which require the maximisation of a probability function, are especially well-suited for this technique for a number of reasons.

Iteratively reweighted least squares is the approach that is applied in the context of robust kernel density estimation. The goal of this algorithm is to optimise the parameters that are related with the Hampel function. In order to reduce the effect that outliers have on the estimate of density, the technique requires making small, incremental changes to the tuning parameters of the Hampel function. The algorithm works by carrying out the following steps:

1. Initialize the turning parameters.
2. Compute the weights for each data point using the current parameters.
3. Update the parameters based on the weighted least squares criterion.
4. Repeat steps 2 and 3 until convergence is achieved.

The Iteratively Reweighted Least Squares method is able to make a dynamic adjustment to the underlying data distribution because of the iterative structure that underpins the methodology. Because of its flexibility, the approach may effectively reduce the impact of data points that are considered to be outliers, while at the same time preserving the precision of its density estimate for data points that are not considered to be outliers.

To summarise, the methodology known as robust Kernel Density Estimation addresses the influence of outliers and data distributions with heavy tails by combining M-estimation with the Hampel function and Iterative Reweighted Least Squares. These two methods are used in conjunction with one another. A persuasive method for attaining accurate density estimate when confronted with challenging data characteristics is presented in the form of the Hampel function. This is owing to the function's ability to deal with extreme values. In the following sections, we study the integration of RKDE, with the support of these robust techniques, into the Naive Bayes classifier in order to increase its performance in the analysis of intelligence data. This is done in order to improve the Naive Bayes classifier's overall accuracy [18].

# Naïve Bayes Classifier

Within the realm of machine learning, the Naive Bayes Classifier is a method for probabilistic classification that is widely used and considered a basic building block of the discipline. The Bayesian probability theory is a resilient framework that is used to handle uncertainty and generate predictions using current data. It is possible that this idea may be traced back to the foundations of Bayesian probability theory.

# Baisc Naïve Bayes Classifier

The underlying premise of Bayes' theorem, which is a key issue within the realm of probability theory, is used as the basis for the construction of the Naive Bayes Classifier. Calculating the probability of an occurrence is made possible by the use of past knowledge about circumstances that have the potential to be related with the event. The use of Bayes' theorem for classification is relevant in the field of NBC, notably in the prediction of the class label () for a given observation based on its attributes (). Bayes' theorem was developed to analyse data in the field of statistics.

Within the realm of mathematics, Bayes' theorem may be formulated in the following way:

* is the posterior probability of class given the features
* is the likelihood of observing features given class
* is the prior probability of class
* is the marginal likelihood of the features

The Naive Bayes Classifier is attractive due to the fact that it is inherently straightforward. The assumption that is being made is that the characteristics are conditionally independent given the class name , which is something that has the potential to be seen as "naive."

In layman's terms, this assumption proposes that knowing the value of one characteristic does not provide any information on the value of another feature, assuming that the class label has already been established. Taking this technique makes the probability word easier to understand.

In this scenario, the variable refers to certain qualities, and the parameter may be estimated with the help of the training data that is now at hand. Estimation methods such as Maximum Likelihood and Kernel Density provide non-parametric alternatives to traditional methods for approximating probability densities. These methods are among the most prominent in the industry [16].

# Challenges with Noisy and Tough Datasets

The Naive Bayes Classifier has been shown to be useful in a variety of applications; but, it does have some limitations, especially when it is presented with datasets that are noisy or that provide a difficulty. The method's vulnerability to outliers and departures from the assumption of independence is a significant limitation that has to be taken into consideration. The assumption of naive independence is called into question when realistic situations are analysed using datasets that are generated from real-world sources. These datasets commonly demonstrate the existence of noisy or connected characteristics. When this assumption is shown to be incorrect, the performance of the Naive Bayes Classifier suffers, perhaps to a significant degree.

Our research is geared towards overcoming these challenges by enhancing the Naive Bayes Classifier with more reliable Kernel Density Estimation methods. This will be our primary focus. Our goal is to increase the robustness and accuracy of the Naive Bayes Classifier by applying robust statistical approaches such as M-estimate with the Hampel function to boost the estimation of . This will allow us to better understand the relationship between the two variables.

In the following parts, we shall investigate the strategy that was used in order to improve the optimisation of RKDE. In order to do this, Harris Hawk Optimisation will be included for the goal of determining bandwidth, and an analysis of the impact these adjustments will have on the performance of classification will be carried out.

This thesis aims to offer a comprehensive examination of the application of RKDE to the Naive Bayes Classifier as its primary research goal. This study intends to improve the classification skills of the Naive Bayes Classifier, especially in circumstances in which the assumptions of data quality and independence are brought into doubt [16].

# Methodology

This section provides an in-depth analysis of the methods that were applied in order to improve the efficiency of the Naive Bayes Classifier. This was accomplished by utilising Robust Kernel Density Estimation in conjunction with Iterative Reweighted Least Squares and Harris Hawk Optimisation for the purpose of determining the bandwidth that provided the best results. When presented with noisy and complicated datasets, which are regularly encountered in the study of intelligence data, the use of these methodologies is of the highest significance in boosting the Naive Bayes Classifier's robustness and predicted accuracy.

# Robust RKDE with IRLS

The following is an explanation of the method that was used throughout the process of adapting and Robust M-estimation to the RKDE:

|  |
| --- |
| **Algorithm: RKDE with IRLS and Robust M-estimation (Hampel function)** |
| **Input:** Training dataset D, Bandwidth parameter h  **Output:** RKDE model |
| 1. Initialization: The process commences by initializing the RKDE model with Gaussian kernel parameters. Additionally, we initialize weights w for each data point in D, which will play a pivotal role in the robustness of the density estimation. |
| 2. Weight Adjustment Iteration: The core of the algorithm consists of iterative weight adjustments. During each iteration, we:  a. Update the weights w: This is achieved using IRLS based on the Hampel function. The Hampel function identifies outliers within the dataset and subsequently down-weights them. This process significantly enhances the robustness of the density estimation.  b. Recalculate the RKDE model: With the updated weights, we recalculate the RKDE model. This model reflects the density distribution while accounting for the identified outliers. |
| 3. Convergence: The iteration continues until convergence is achieved. |
| 4. **Return:** The final RKDE model is returned, which encapsulates the robust density estimation capable of handling challenging datasets. |

This algorithm offers a full examination of the computational complexity involved in effectively predicting the Kernel Density by combining the Iteratively Reweighted Least Squares technique with the RKDE approach. The technique relies on an iterative process, during which the Hampel function is used to make adjustments to the weights of the individual data points. The Hampel function plays a significant part in the process of identifying and then reducing the number of outliers, which ultimately results in an improvement in the robustness of the density estimation procedure [17,18,19].

# Harris Hawk Optimization (HHO) for bandwidth selection

# Harris Hawk Optimization

In the realm of computational intelligence, the swarm intelligence algorithm is a method that is used extensively, and in the area of evolutionary computing, it is a methodology that is still in the process of evolving. The primary idea behind it is to simulate the actions of a wide variety of species that are found in nature, such as ants, birds, bees, wolves, germs, and so on. Swarm intelligence is an approach to solving complex issues that seeks to tap into the collective intellect of many groups by capitalising on the mechanisms that allow for interaction and information sharing among them. The topic of intelligent algorithms has been the subject of extensive study by a large number of academics, which has led to the development of a wide variety of cutting-edge algorithm designs. The grey wolf optimizer (GWO), the lightning search algorithm (LSA), the marine predators algorithm (MPA), the sine cosine algorithm (SCA), the salp swarm algorithm (SSA), the water cycle algorithm (WCA), the cuckoo search (CS), the artificial bee colony (ABC), and the moth flame optimizer (MFO) are some of these algorithms. The Harris Hawks Optimisation (HHO) approach is a very new swarm intelligence technology that was only recently designed and launched. The conduct of Harris hawks as they interacted with their prey, which included both pursuit and avoidance strategies, served as the impetus for the development of this algorithm. Since its creation, the simplicity of the algorithm in theory, together with its small number of parameters and powerful global search capability, has attracted substantial attention and extensive application across a broad variety of engineering fields. However, like to other intelligent optimisation algorithms, the basic Harris hawks method is prone to various limitations. These constraints include a decreased degree of precision in obtaining convergence and a propensity to get stranded in local optima when trying to tackle sophisticated optimisation problems. [33]

By using a wide variety of different optimisation tactics, the HHO algorithm is able to significantly improve both the optimisation capabilities of local exploitation and those of global exploration. This is the major improvement that the HHO algorithm offers. Because of this, the algorithm's convergence accuracy and overall performance are both enhanced, making it more suited for use in actual engineering applications.

Kernel density estimation is a method of machine learning that may be successfully improved by the use of a powerful tool for parameter optimisation. This can be accomplished in a number of different ways [36].

# Description HHO

The HHO algorithm is a mathematical formulation that was designed to imitate the hunting strategy used by Harris hawks. Within the context of this algorithm, individual Harris hawks stand in for potential answers, while the prey is meant to stand in for the best possible answer attained after each iteration. The algorithm is broken down into its basic components, which are the exploration stage and the exploitation stage. The amount of energy that the prey has to flee is the primary factor that decides when these phases shift into one another. The first iteration of Harris and Hawks' optimisation algorithm is going to be broken down and discussed in the next section.

* **Exploration phase**

The global search phase is mostly influenced by the geographical data pertaining to the population of Harris hawks, and its update approach can be described as follows:

The spatial coordinates of the hawks in the iteration 𝑡+1 are denoted by the variable 𝑋(𝑡+1) in the context that has been presented. The spatial coordinates of the prey are denoted by the variable 𝑋𝑝𝑟𝑒𝑦(𝑡). The value of 𝑋(𝑡) represents the spatial location of the hawks in the generation t that is now active. The values for the variables 𝑟1–𝑟4 and q are created at random within the range of (0, 1), and these values are refreshed at the beginning of each iteration. Both the upper and lower boundaries of the population are denoted by the notations 𝑈𝐵 and 𝐿𝐵, respectively. 𝑋𝑟𝑎𝑛𝑑(𝑡) is the name given to a hawk that was chosen at random from the present population. The equation used to determine the average spatial location of individuals in the current population is denoted by the symbol 𝑋𝑚(𝑡) , and it is called Equation (9).

Where: represents the positional value of hawk at iteration , whereas n represents the total number of hawks.

* **Transition from Exploration to Exploitation**

The equation governing the escape of prey is represented by the energy equation.

Where: represent the current iteration count, represent the maximum number of iterations, and symbolise a randomly generated number within the range of (-1,1) that signifies the starting energy state. When the magnitude of the escape energy |𝐸| is greater than or equal to 1, Harris hawks engage in a global exploration phase by searching various locations in order to locate the prey. On the other hand, when the magnitude of |𝐸| is less than 1, Harris hawks engage in a local exploitation phase by exploring nearby solutions.

**- Exploitation Phase**

In this step, the Harris hawk will engage in a predatory conduct against its intended prey upon identifying it. This behaviour will draw upon the discoveries of the stages that came before it. Concurrently, the target of the chase will make an effort to elude its pursuers. In this part of the simulation, depending on the behaviours of Harris's hawks and their prey, there are four different strategies that may be used. Both the aggressive and passive besieging actions that were modelled after Harris's hawk are displayed by E. The value of the “r” parameter is supposed to indicate the success or failure of the prey's effort to flee.

1. **Soft Beseige**

In situations when the absolute value of the energy parameter (|𝐸|) is more than or equal to 0.5, and the distance parameter (𝑟) is greater than or equal to 0.5, the prey organism exhibits a behavioural response of attempting to evade the predator by means of jumping. Concurrently, the Harris hawk predator employs a strategy of employing a gentle and steady approach to deplete the prey organism's energy reserves. The behaviour is represented in the following manner:

𝑋(𝑡+1)=Δ𝑋(*𝑡*)−𝐸|𝐽𝑋𝑝𝑟𝑒𝑦(*𝑡*)−𝑋(𝑡)| Eq.(11)

Δ𝑋(𝑡)=𝑋𝑝𝑟𝑒𝑦(𝑡)−𝑋(𝑡) Eq.(12)

Where 𝑟5 is used to denote a number that is created randomly from a uniform distribution between 0 and 1. The variable J is incorporated into the model to mimic the characteristics of prey movement, and its value is subject to random variation in each iteration.

1. **Hard Beseige**

When the absolute value of E is less than 0.5 and r is greater than or equal to 0.5, the prey lacks the energy to flee. Consequently, the Harris hawk engages in an aggressive siege approach, employing Equation (13) to modify its present location.

𝑋(𝑡+1)=𝑋𝑝𝑟𝑒𝑦(*𝑡*)−𝐸|Δ𝑋(*𝑡*)| Eq.(13)

1. **Soft Beseige with Progressive Rapid Dives**

When the absolute value of 𝐸 is greater than or equal to 0.5 and 𝑟 is less than 0.5, it can be concluded that the prey possesses sufficient energy to successfully evade the predator. The Harris hawks will adjust their locations in accordance with the governing principle outlined in Equation (14):

𝑌=𝑋𝑝𝑟𝑒𝑦(𝑡)−𝐸|𝐽𝑋𝑝𝑟𝑒𝑦(𝑡)−𝑋(𝑡)| Eq.(14)

𝑍=𝑌+𝑆×𝐿𝐹(𝐷) Eq.(15)

In the context of the problem, *D* represents the dimensionality of the system under consideration. *S* denotes a random vector with dimensions 1*×D*. Additionally, *LF* refers to the levy flight function, which may be mathematically expressed as shown in Equation (16).

where u and v are random values within (0,1) and 𝛽 is the default constant, set to 1.5.

Therefore, the implementation of Equation (17) can serve as the ultimate approach for updating the positions of the hawks during the soft siege phase.

where Y and Z are obtained using Equations (11) and (12), respectively.

1. **Hard Beseige with Progressive Rapid Dives**

When the absolute value of E is less than 0.5 and r is less than 0.5, it can be concluded that the prey lacks sufficient energy to successfully escape. In such circumstances, a specific strategy is established to be implemented.

𝑌=𝑋𝑝𝑟𝑒𝑦(𝑡)−𝐸|𝐽𝑋𝑝𝑟𝑒𝑦(𝑡)−𝑋𝑚(𝑡)| *Eq.(19)*

𝑍=𝑌+𝑆×𝐿𝐹(𝐷) Eq.(20)

where is obtained using Equation (9) [37].

# The Process of HHO

The main steps of the overall HHO algorithm are as shown in the following algorithm.

|  |
| --- |
| **Algorithm : Main steps of HHO algorithm** |
| **Input**: Population size N and the maximum number of iterations *T* |
| 1: Initialize the population  2: **while** 𝑡<𝑇 **do**  3: Calculate the fitness of each solution and get the optimal individual  4: **for** i=1:N **do**  5: According to Equation (3) update the escape energy E  6: **if** |𝐸|≥1 **then**  7: According to Equation (1) update the location  8: **else** **if** 𝐭𝐡𝐞𝐧|𝐸|<1  9: **if** |𝐸|≥0.5 and 𝑟≥0.5 **then**  10: According to Equation (4) update the location  11: **else if** 𝐭𝐡𝐞𝐧|𝐸|<0.5 and 𝑟≥0.5  12: According to Equation (6) update the location  13: **else** **if** 𝐭𝐡𝐞𝐧|𝐸|≥0.5 and 𝑟<0.5  14: According to Equation (10) update the location  15: **else if** 𝐭𝐡𝐞𝐧|𝐸|<0.5 and 𝑟<0.5  16: According to Equation (11) update the location  17: **end if**  18: **end if**  19: **end for**  20: 𝑡= 𝑡+1  21:**end while**  22: **return** 𝑋𝑝𝑟𝑒𝑦 |

# Selecting Bandwidth with HHO

The HHO algorithm is a modern evolutionary optimisation method that takes its cues from the hunting techniques used by Harris hawks as a source of inspiration. The method of kernel density estimation is one of the machine learning processes that may be considerably improved by making use of a powerful instrument for optimising the parameters of the model.

|  |
| --- |
| **Algorithm : Harris Hawk Optimization for Bandwidth Selection** |
| **Input**: RKDE model, Dataset *D*  **Output**: Optimized bandwidth parameter *h* |
| 1. Initialization: HHO begins by initializing a population of hawks with random positions. |
| 2. Fitness Evaluation: The fitness of each hawk is evaluated based on the performance of the RKDE model using Cross-Validation. This fitness assessment guides the search for an optimal bandwidth. |
| 3. Leader Identification: The algorithm identifies the best-performing hawk (leader) in the population, based on fitness. |
| 4. Iteration: The optimization process unfolds over a predefined number of generations. Within each generation, the following steps are executed:  a. Position Update: The positions of hawks are updated using specific position update rules.  b. Fitness Re-evaluation: After position updates, the fitness of each hawk is re-evaluated.  c. Leader Update: The leader hawk is updated based on fitness. This leader guides the exploration of the solution space.  d. Exploration and Exploitation: The algorithm fine-tunes its exploration and exploitation parameters based on the observed fitness landscape. |
| 5. Result: The optimized bandwidth parameter h is determined based on the leader hawk's findings throughout the optimization process. |

An in-depth explanation of the use of Harris Hawk Optimisation to RKDE's bandwidth parameter optimisation is provided in the method. This is done with the intention of achieving maximum performance. The RKDE model's robustness and prediction accuracy are both significantly improved as a result of the suggested algorithm's methodical exploration of the solution space, which aims to find the optimum bandwidth at which to operate the model.

This section discusses the key tactics that were used in this work to alter RKDE and the Naive Bayes classifier in order to effectively handle datasets that are noisy and demanding. These modifications were made in order to improve accuracy. The inclusion of these methods plays a crucial role in meeting the purposes of the thesis, which seek to achieve reliable and precise categorization in the study of intelligence data. The thesis attempts to get reliable and precise categorization by using the methods described in the previous sentence. In the next sections, we are going to go more deeply into the experimental setup, results, and repercussions of these different techniques.

# Complete Cross-Validation Curve (CCV)

In this section, the approach known as the Complete Cross-Validation Curve is discussed. This methodology is an important part of our study since it permits the selection of bandwidth while doing the Robust Kernel Density Estimation. Bootstrap, Unbiased Cross-Validation, and Biassed Cross-Validation are the three main components that make up the Complete Cross-Validation approach. In the context of RKDE, each individual component plays a significant role in strengthening the robustness and accuracy of bandwidth selection.

# Unbiased Cross-Validation

The methodology known as Unbiased Cross-Validation, which is also sometimes called Leave-One-Out Cross-Validation (LOOCV), serves as the foundation for CCV. The method entails removing individual data points in an iterative manner and evaluating the model's performance in comparison to the data that is still there in order to determine how successful it is. The following is the mathematical description of the process being described:

Where: *n* is the number of data points, *K(·)* is the kernel function, and are data points, *h* is the bandwidth, a parameter to be selected.

The BCV bandwidth selector was originally introduced by Scott and Terrell in 1987. Jones and Kappenman (1992) provided a theoretical analysis comparing the Local Scoring Cross-Validation (LSCV), Biased Cross-Validation and other bandwidth selections.

* **Strengths**

The capability of Unbiased Cross-Validation, often known as UCV, to produce estimates of model performance that are free from bias is one of the method's most prominent benefits.

Evaluation Free from Prejudice: The use of Unbiased Cross-Validation (UCV) makes it possible to conduct an objective analysis of the performance of the model. This is accomplished by consciously excluding some data points from the study and determining the model's ability to forecast using just the data that was not removed from consideration. This procedure ensures that all students are evaluated fairly.

Because it is based on solid mathematical ideas, the UCV methodology provides a reliable tool for assessing the effectiveness of various models.

The approach has a low processing cost since, in contrast to other cross-validation procedures, it requires a much lower number of computations to be performed. Because to this property, it has a high level of computing efficiency, which is especially useful when working with large datasets.

* **Weaknesses**

High degree of uncertainty The Unbiased Cross-Validation approach has the potential to exhibit a high degree of uncertainty, especially in circumstances in which the dataset is of a small size. Because some data points are left out of consideration throughout each iteration, the results may be subject to a significant amount of variation.

It is possible for the Unweighted Covariance approach to display sensitivity to outliers since it assigns the same weight to each data point throughout the assessment phase. This causes the method to lack robustness to outliers. The presence of outliers has the potential to have an uneven and disproportionate impact on the results.

# Biased Cross-Validation (BCV)

In order to account for the possible biases that result from the iterative elimination of individual data points, the Biased Cross-Validation approach integrates bias correction into the UCV methodology. This is done in order to address the issue of potential biases. The following is an expression in mathematics that represents BCV:

where: *n* is the number of data points, *K(·)* is the kernel func- tion, *Xi* and *Xj* are data points, *h* is the bandwidth, a parameter to be selected.

* **Strengths**

The ability of BCV to generate reliable estimates of model performance is one of its primary features. This is especially useful in circumstances in which the training data is either restricted or unbalanced. This is accomplished by BCV by the deliberate introduction of bias into the cross.

The issue of large variance that is associated with UCV is addressed by BCV via the use of a process known as bias correction. When used, this method produces assessments of the model's performance that are more stable and subject to less variation when compared to other methods.

The capacity of BCV to take into account outliers contributes to the model's superior resilience compared to that of UCV. BCV accomplishes this goal by methodically removing individual data points and accounting for any possible biases that may be introduced as a result.

A Fair and Balanced Compromise: The BCV approach creates a harmonic equilibrium between bias and variance, making it well-suited for datasets that display varying quantities of noise and outliers due to the method's ability to strike a balance between the two.

* **Weaknesses**

One of the drawbacks of using a method known as biassed cross-validation is the fact that it is susceptible to bias. Due to the extra processing resources necessary for the additional computations that are involved in bias correction, the computational cost of BCV is greater than that of UCV. This is because the bias correction procedure involves more calculations.

# Bootstrap

In order to strengthen the validity of the CCV, the bootstrap approach, along with the UCV and BCV, is often used. The procedure comprises producing several resampled datasets by randomly selecting data points and then replacing those points with new ones. Following that, the aforementioned datasets are put to use in order to calculate the CCV scores based on a variety of bandwidth characteristics. The mathematical expression that may be used to describe Bootstrap Aggregating (Bagging) is as follows:

Where:

* signifies the BCV score obtained through Bootstrap Aggregating for a given bandwidth parameter ℎ
* represents the BCV score calculated for the *b-th* bootstrap dataset.
* *B* denotes the number of bootstrap datasets.

The Bootstrap method may make significant demands on a computer's resources, especially in terms of the amount of time it takes to do computations, when it is used to generate a large number of datasets that have been resampled. When dealing with very large datasets, the aforementioned issue may provide a challenge that has to be addressed and resolved. There is a chance that the utility of Bootstrap is contingent on the selection of variables, one of which is the quantity of bootstrap samples utilized, which might lead to the introduction of subjectivity. This possibility exists because there is a possibility that the usefulness of Bootstrap is reliant on the selection of factors.

These component characteristics are incorporated into the CCV framework in order to make the most of their one-of-a-kind benefits and to relieve the inherent limits that are fundamental to these parts. The preliminary estimate is provided by the UCV, which serves as a jumping off point for additional inquiry. After that, the BCV technique is used in order to address any potential biases that may be present in the data. This is done so in order to make the data as accurate as possible. In conclusion, the Bootstrap approach is used in order to further strengthen the outcomes' trustworthiness as well as their resilience. It is particularly well-suited for the analysis of intelligence data, which usually consists of noisy and complicated datasets, and the use of this combination provides a full review of bandwidth selection in RKDE.

# Importance of using CCV in RKDE and Intelligence Data Analysis

The usage of CCV has major relevance in the area of RKDE and intelligence data analysis because to the inherent qualities of datasets, which often display noise, complexity, and sensitivity to outliers. This is because of the inherent characteristics of datasets. There are many different explanations for this:

**Comprehensive Assessment**: The CCV performs an in-depth analysis of the RKDE model's performance in order to provide an accurate assessment. UCV, which stands for unbiased cross-validation, is used in the approach to conduct objective evaluations. BCV, which stands for bias corrective validation, is used to compensate for bias, while Bootstrap is used to make the methodology more resilient. This ensures that the chosen bandwidth is capable of effectively handling datasets that are noisy and present issues in the appropriate manner.

**Mitigating Limitations**: The CCV successfully solves the limitations that are inherent in each individual element by integrating a variety of approaches. When used in this setting, the practice of Bootstrap assists to reduce the inherent high variance associated with UCV, so addressing the possibility of bias and fostering a more stable environment.

**Robust Decision-Making**: In order to successfully complete the task of assessing collected intelligence data, one must use effective decision-making methods. A detailed technique was employed by CCV in the selection of bandwidth for RKDE. This methodology assures a rigorous examination of the model's effectiveness, which establishes the model's dependability in real-world applications.

In conclusion, the use of the CCV approach shows itself to be a very useful instrument that may be found in the toolkit of intelligence data analysts. In the context of RKDE, the suggested technique provides a strategy that is exhaustive and exhaustive in its approach to identifying the optimal bandwidth. This strategy, in the long run, improves the dependability and resilience of modelling results, especially when working with complicated datasets.

# Integration of Robust KDE and Bandwith selection into Naïve Bayes classifier/ Enhancing Naïve Bayes Classifier with Robust KDE and Bandwidth Optimization / Robust Kernel Density Estimation and Bandwidth Selection in Naïve Bayes Classification

A significant step forward in the field of intelligence data processing has been marked by the Naive Bayes classifier's adoption of the RKDE algorithm and bandwidth selection as two of its key components. The integration that is provided in this study seeks to address the inherent challenges that are connected with datasets that are noisy, intricate, and likely to include outliers. The Naive Bayes classifier is able to significantly improve both the accuracy and the robustness of its predictions by using this integration.

In order to successfully complete this integration, a series of algorithmic steps are carried out in the order that is outlined in the following manner:

* Step 1: Data Preprocessing

It is required to do preprocessing on the dataset before integrating it in order to resolve any instances of missing values, outliers, and other problems connected to the quality of the data. This must be done before the integration can take place. This step is very necessary in order to guarantee that the data will be usable for the subsequent steps in the process.

* Step 2: Robust KDE

The first stage is estimating the PDF of the data by using RKDE. PDF is a crucial component inside the framework of the Naïve Bayes Classifier. The input consists of Dataset D, Hampel function H . The HHO approach is used to optimize the bandwidth parameter for the RKDE. The choice of bandwidth is a critical factor in KDE, and HHO aids in the determination of an optimal bandwidth value that enhances the robustness of the RKDE.

The global of our RKDE is defined as:

|  |
| --- |
| **Algorithm : RKDE with IRLS and Robust M-estimation (Hampel function)** |
| **Input:** Data *X*, Number of iterations *N*  **Output:** Robust Kernel Density Estimate*p(·)*  **Initialize:** *w* **←** ones(*n***)** Initialize weights to ones |
| 1: Find optimal bandwidth σ Apply HHO algorithm  2: **for** *i*←1 to *N* **do**  3: Compute initial KDE using Gaussian kernel and bandwidth σ  4: Compute residuals  5: Compute robust weights *w* = *hampel*(*r*, δ) Apply Hampel algorithm  6: Compute new KDE *p(·)* using weighted kernel and updated bandwidth  7: Compute new residuals *r = X − p(X)*  8: Compute updated weights *w* = *hampel*(*r*, δ) Apply Hampel algorithm  9: **endfor**  22: **return** Robust Kernel Density Estimate *p(X)* |

* Step 4: Feature Selection (if applicable)

It is possible that, in some circumstances, it will be beneficial to participate in feature selection as a method of further enhancing the performance of categorization. During this part of the process, you will be tasked with identifying the properties of the dataset that are most important.

* Step 5: Naïve Bayes Classifier with RKDE

We combine these components into the Naive Bayes Classifier, including the robust probability density estimate that was produced via the RKDE approach as well as the optimal bandwidth. The adoption of this strategy considerably increases the classifier's potential to draw well-informed conclusions, particularly when faced with outliers and noisy data. This is especially the case when the classifier is asked to analyze the data.

The incorporation of the RKDE technique and the process of bandwidth selection into the Naïve Bayes Classifier has several benefits:

**Robustness:** by giving a more accurate prediction of the underlying data distribution, the use of the RKDE approach makes our classifier more resistant to outliers and noisy data. This is accomplished via the utilization of the Robust Kernel Density Estimation.

**Optimal Bandwidth:** the use of HHO makes it possible to optimise the bandwidth parameter, which ultimately leads to an improvement in the performance of the KDE approach.

**Enhanced Classification:** The Naïve Bayes Classifier exhibits enhanced classification efficacy, especially when confronted with intricate datasets, owing to the use of a more accurate probability density estimate.

**Data-Driven:** the integration process is led by the data, which enables it to adapt to and accommodate the particular characteristics of the dataset. As a result, the adaptability of the dataset is increased so that it may be used in a wider variety of contexts.

In the parts that follow, an in-depth analysis of the experimental design, the results, and the implications of this integration will be provided.

# Experimental Setup

# Dataset Description

This section includes a full explanation of the datasets that were used for the purpose of testing the performance of the proposed approach, which is incorporated into the Naive Bayes classification framework. In addition to genuine data collected from the actual world, the datasets include artificial data organised in a two-dimensional space and produced using a variety of distributions. When attempting to evaluate the adaptability and performance of integrating RKDE-Naive Bayes in a variety of contexts, it is of the highest necessity to pick datasets with extreme care.

In all of the research, the kernel function has been determined to be the Gaussian kernel. Due to the intrinsic smoothness of the Gaussian kernel as well as the mathematical tractability of the problem, it is often chosen for use in the process of density estimation. The bandwidth parameter, denoted by the Greek letter, is an extremely important component of the Gaussian kernel. A variety of different optimisation approaches, which include the following, are utilised in order to determine the bandwidth that is the most appropriate.

* Particle Swarm Optimization (PSO) [27, 28]: PSO is a technique of optimization that takes its cues from the collective behavior that may be seen in populations of fish or birds. In order to find the bandwidth that allows for the RKDE technique to operate at its highest level of efficiency, a methodological approach is performed.
* Black Widow Optimization (BWO) [29]: in the context of RKDE, the BWO method is an innovative and one-of-a-kind metaheuristic optimization methodology that is used for the goal of improving the bandwidth of the Gaussian kernel.
* Quantum Particle Swarm Optimization (QPSO) [32]: in order to effectively improve the bandwidth parameter, the QPSO method combines the core ideas of quantum computing with the Particle Swarm Optimisation (PSO) approach. This integration was carried out in order to achieve optimal performance.

Utilising Hampel's *H(x)* function is an integral part of the procedure that determines which values are assigned to the tuning parameters *a, b,* and *c* in the CCV objective function. The variable *"a"* is used to represent the median absolute deviation of the data, while the variables *"b"* and *"c"* are used to denote, respectively, the *75th* and *95th* percentiles of the distribution of the data. When it comes to optimizing the bandwidth of the Gaussian kernel and, as a result, increasing the robustness of the RKDE-Naive Bayes classifier, the tuning parameters are of the highest significance.

# Synthetic data

In order to determine how reliable the RKDE approach is, we put it through a battery of tests using three different sample sizes of two-dimensional data: 2,000, 3,000, and 4,000 respectively. The data comes from a variety of distributions, including a Gaussian distribution (Equation 25), a uniform distribution (Equation 24), and other distributions. The production of synthetic data takes place within a range that has been determined in advance. The primary dataset in each distribution is denoted by the letter *F0*, whereas the dataset that contains contaminated data is denoted by the letter *F1*. It is important to note that the fundamental frequency (*F1*) is responsible for *10%* (can be change) of the overall fundamental frequency (*F0*). This is something that should be mentioned.

where denotes the uniform distribution on the interval [*a, b*].

*x* represents the variable, *μ* is the mean, and *σ* is the standard deviation of the Gaussian distribution.

# Real-World data

In addition, we evaluate the effectiveness of the RKDE-Naive Bayes classifier by employing datasets taken from the actual world, in addition to data that was created in a lab. When analysing data from the actual world, researchers often run across challenges such as the existence of noise, outliers, and distributions with varying degrees of similarity. The addition of actual data into our evaluation provides vital insights into the practical usability and robustness of the classifier.

Given that the empirical Banana dataset displays intricate, non-linear patterns, it is an excellent option for evaluating the robustness of the Naive Bayes approach due to its suitability in this context. A total of 5,300 samples are included in the dataset that is being studied, which is a two-dimensional dataset. The structure that was produced as a consequence has a pattern that can be recognised by the existence of two clusters that cross one another and have a form that is similar to that of bananas. As a direct consequence of this, conventional techniques for estimating density have difficulties when it comes to correctly portraying the underlying range of densities.

# Implementation Details

The success of an implementation is based on a number of critical components and settings being properly applied.Python is the programming language that serves as the foundation for the software architecture that houses the RKDE-Naive Bayes classifier. Python is a programming language that is widely used in the fields of machine learning and data science. It provides access to a wide variety of libraries and tools, all of which are required for our implementation. NumPy's features allow us to successfully handle data and carry out numerical operations, both of which are essential to the job that we do. In addition, we make use of SciPy for tasks associated with scientific computing, and Scikit-learn for the construction and evaluation of machine learning models. Both of these programmes are written in Python.Within the context of RKDE, the Gaussian kernel functions as an essential component for density estimation. The Rosenblatt kernel density estimation (RKDE) approach relies on the accurate determination of an ideal bandwidth (*h*), making it very essential that this variable be chosen correctly. In order to make the process of selection as efficient as possible, a variety of cutting-edge algorithms for choosing bandwidth have been implemented.

# Performance metrics

We make use of a broad variety of performance measures in order to determine how successful the RKDE-Naive Bayes classifier that we have provided is. These metrics provide vital insight into the classifier's ability to effectively manage datasets that are both noisy and demanding in nature.

# Accuracy

Accuracy is one of the most important parameters to consider when assessing the performance of a classifier. The ratio of correctly recognised instances to the total number of occurrences in the dataset is what this statistic attempts to quantify and assess. It is possible to phrase it in the following way from a mathematical point of view:

Where:

(True Positives) represents the number of correctly classified positive instances.

(True Negatives) represents the number of correctly classified negative instances.

(False Positives) represents the number of negative instances classified as positive.

(False Negatives) represents the number of positive instances classified as negative.

# Precision

The accuracy of positive classifications is the primary focus of the precision metric as it was designed. The idea of a ratio, more precisely the ratio of true positive classifications to the total number of positive classifications, is what is meant when "it" is mentioned.

# Recall

The ability of the classifier to reliably recognize positive occurrences may be measured using the metric of recall, which is also frequently referred to as sensitivity or the true positive rate. In order to complete the computation, you will need to determine the percentage of correctly classified positive instances in relation to the total number of cases that really are positive.

# F1-Score

The F1-Score is a mathematical formula that reflects the harmonic mean of a participant's accuracy and recall scores. It is a statistic that is well-balanced and serves the purpose of determining the effectiveness of a classifier.

# Area Under the Curve (AUC)

The AUC provides a comprehensive measure of the classifier's overall performance. A greater AUC is indicative of superior discriminative capacity. It is one of the most important metrics for evaluating the efficacy of any classification model.

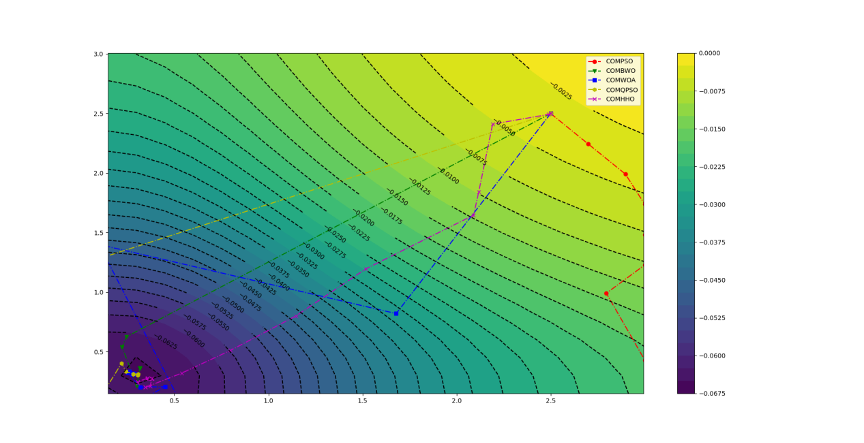
In the following sections, we will show the outcomes and comments obtained from our experiments. We will use performance measures to evaluate the efficacy of the RKDE-Naïve Bayes classifier in dealing with both synthetic and real-world datasets.

# Results and Discussion

In this section, we present the results of our experiments and delve into a comprehensive discussion of the findings. The evaluation of the Optimized RKDE in the context of Naïve Bayes classification, together with the influence of bandwidth selection on classification performance, forms the crux of our analysis. Additionally, we compare the performance of our approach with that of the traditional Naïve Bayes classifier.

# Evaluation of Optimized Robust KDE

The experimental results suggest that the bandwidth obtained from the CCV results exhibit superior performance compared to the MISE values of Bootstrap, UCV, and BCV. Based on this discovery, it appears that CCV manufactures window sizes that are more appropriate. Figure 1 depicts the influence of the goal function in situations with two dimensions.



**Fig 1:** CCV objective function with x- and y-axis indicating the window width in both dimensions h

The results of the experiments are summarised in Table 1 and 2, and the optimal bandwidth is then derived and evaluated for optimisation in combination with robust kernel density estimation.

|  |  |
| --- | --- |
| **Objective functions** | **MISE** |
| BCV | 7.40144418E-06 |
| UCV | 6.59422965E-06 |
| Bootstrap | 1.75642284E-06 |
| CCV | 1.58630E-04 |

**Table 1:** Final MISE calculated for window width h obtained with different objective functions

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objective**  **Functions** | **PSO** | **BWO** | **QPSO** | **HHO** |
| BCV | 2.4E-04 | 1.7E-06 | 3.2E-06 | 2.0E-06 |
| UCV | 1.7E-05 | 5.7E-06 | 7.4E-06 | 6.7E-06 |
| Bootstrap | 2.5E-04 | 5.5E-06 | 2.5E-04 | 2.5E-04 |
| CCV | 1.59E-04 | 2.6E-06 | 2.0E-06 | 2.15E-06 |

**Table 2:** Objective functions - MISE

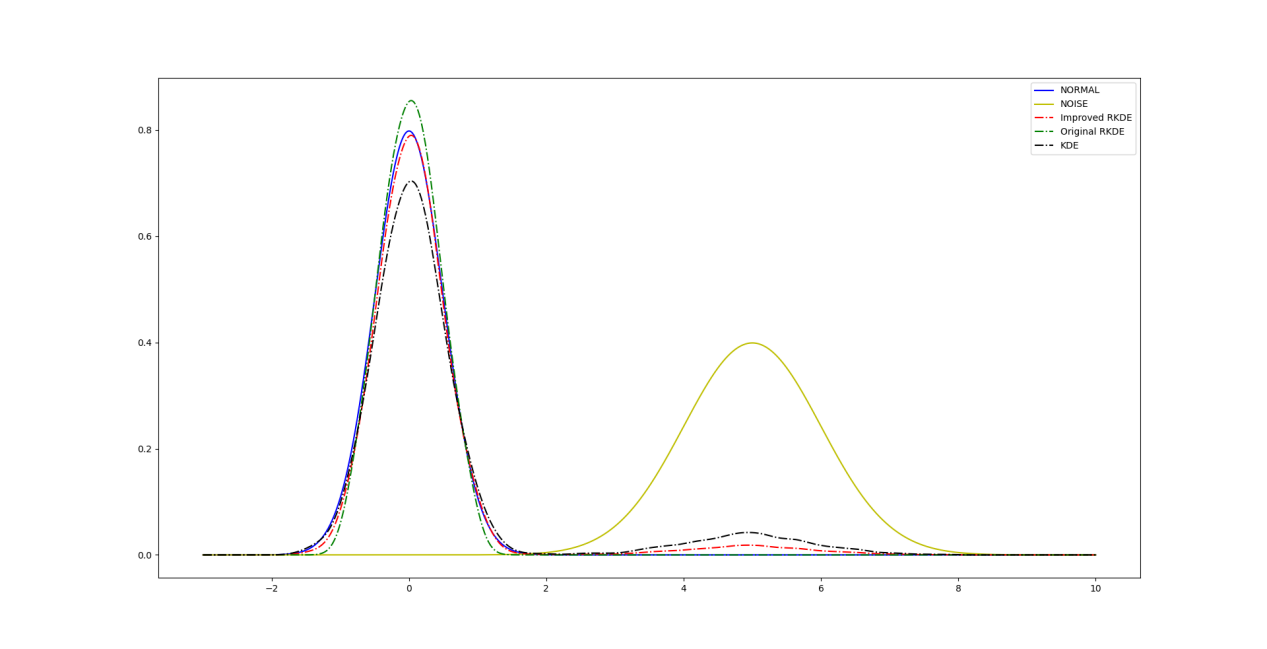
The MISE comparison for several two-dimensional distribution datasets is shown in Table 3, after optimisation through robust kernel density estimation and window width change in the presence of polluted data. This optimisation was carried out in order to improve performance.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Distribution  MISE/Data  volume | **F0=2000** | **F0=3000** | **F0=4000** | **F0=5000** | **F0=6000** |
| Gaussian  distribution  F0+F1 | 0.00027542 | 0.00025047 | 0.00027389 | 0.00026813 | 0.00025792 |
| Gaussian  distribution  hampel  F0+F1 | 0.00019253 | 0.00021888 | 0.0002134 | 0.00023998 | 0.00025938 |
| Gaussian distribution nhampel F0+F1 | 0.000069265 | 0.000018924 | 0.00014733 | 0.00013889 | 0.000013826 |
| Gamma distribution F0+F1 | 0.00058395 | 0.00051061 | 0.00046789 | 0.00046473 | 0.00044839 |
| Gamma distribution hampel F0+F1 | 0.00101694 | 0.00097002 | 0.00114537 | 0.00118356 | 0.00120638 |
| Gamma distribution nhampel F0+F1 | 0.00040906 | 0.00032493 | 0.0001264 | 0.00028466 | 0.00024926 |
| T distribution F0+F1 | 0.00038516 | 0.0004036 | 0.00039858 | 0.00037383 | 0.0003568 |
| T distribution hampel F0+F1 | 0.00051981 | 0.00054315 | 0.00064179 | 0.00061748 | 0.00063738 |
| T distribution nhampel F0+F1 | 0.000068856 | 0.000086692 | 0.0001035 | 0.000064889 | 0.000068181 |
| Weber distribution F0+F1 | 0.00138714 | 0.00126648 | 0.00111878 | 0.00104416 | 0.00103001 |
| Weber distribution hampel | 0.00036791 | 0.00054808 | 0.00051849 | 0.00066513 | 0.00067683 |
| Weber distribution nhampel F0+F1 | 0.00031832 | 0.0002371 | 0.00016147 | 0.00012838 | 0.00012036 |
| Cardinal distribution F0+F1 | 0.00019803 | 0.00017977 | 0.00016904 | 0.00017337 | 0.00016978 |
| Cardinal hampelF0+F1 | 0.00054732 | 0.0004484 | 0.0004804 | 0.00053032 | 0.00053684 |
| Cardinal nhampelF0+F1 | 0.00016426 | 0.00015252 | 0.00014412 | 0.0001502 | 0.00014702 |

**Table 3:** MISE for different distribution data

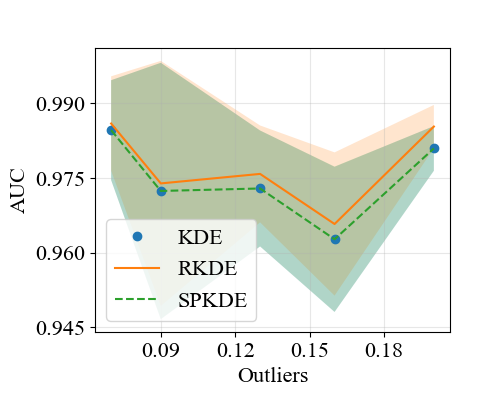
The findings that were reported in Table 3 indicate that the improved RKDE approach displays a low MISE, which indicates its effectiveness in reducing the impact of outliers and generating a probability density estimate that closely approximates the true distribution of the data. The findings also indicate that the improved RKDE approach displays a low MISE.

Figure 2 presents a plot that depicts a dataset that has 2000 sample data points and an extra 200 noise samples. This dataset was used to show the plot. It is obvious that the RKDE face has a greater suppression impact on the polluted data, and there is a very large change in the KDE curves in the low-density zone. This is the case because the RKDE face has a higher KDE value.



**Fig 2:** Improved RKDE

The AUC, which is a technique for measuring performance, was used in a comparison study that was carried out with the purpose of determining the quantitative differences that exist between KDE and RKDE. According to the findings of the observation, the AUC values that were determined using RKDE were less influenced by the existence of outliers when compared to the AUC values that were generated using KDE. This indicates that RKDE performed better than KDE in this respect. Figure 3 illustrates this point well.

****

**Fig 3:** Comparison between RKDE and KDE with uniform

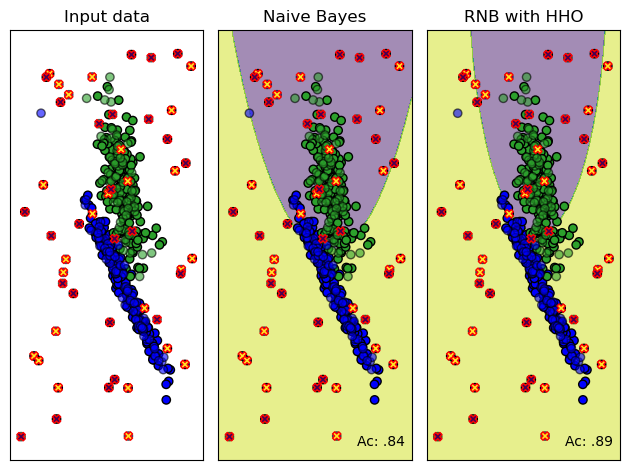
Distribution

# Comparison with traditional Naïve Bayes Classifier

We undertake a comparison study between our suggested Optimised RKDE-Naive Bayes classifier and the conventional Naive Bayes classifier in order to evaluate the degree to which the latter is preferable than the former. For the purpose of this comparison, both synthetic and real-world datasets will be used. We do an in-depth analysis of the performance of both classifiers in terms of classification, and then we compare and contrast the two sets of findings. We demonstrate the benefits of using RKDE as a preprocessing step for Naive Bayes classification by comparing accuracy, precision, recall, and F1-score. The comparative study demonstrates the potential of our methodology in the context of dealing with noisy and complex information.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Dataset (F0 / F1)** | **Method** | **Precision** | **Recall** | **F-1** | **Accuracy** | **Computation**  **(second)** |
| Make\_circle  (1500 / 0.1) | NB | 0.867 | 0.813 | 0.839 | 0.833 | 0.7 |
| Robust NB | 0.843 | 0.875 | 0.859 | 0.84 | 4.2 |
| Make\_moon (1500 / 0.15) | NB | 0.808 | 0.797 | 0.803 | 0.807 | 0.7 |
| Robust NB | 0.817 | 0.905 | 0.859 | 0.853 | 4.6 |
| Uniform  (2000 / 0.2) | NB | 0.84 | 0.87 | 0.85 | 0.84 | 0.8 |
| Robust NB | 0.83 | 0.98 | 0.88 | 0.89 | 7.7 |
| Make\_classification (2000 / 0.1) | NB | 0.82 | 0.78 | 0.8 | 0.79 | 0.9 |
| Robust NB | 0.79 | 0.89 | 0.85 | 0.83 | 7.1 |

**Table 4:** Comparison between Naive Bayes and Robust Naïve Bayes (with RKDE)



**Fig 4:** Comparison between Naive Bayes and Robust Naïve Bayes (with RKDE), on 1000 synthetic data with 20 % of proportion of outliers

According to the data shown in Table 4 and Figure 4, the combination of RKDE and the traditional Naive Bayes classifier led to a significant increase in the accuracy of classification, as was seen in this research.

The experimental research of real data set entailed dividing the Banana dataset into a training set, which contained 70% of the data, and a test set, which comprised the remaining 30% of the data. This was done as part of the Banana dataset.On the training data, the RKDE-based density estimate is employed in conjunction with the Naive Bayes classifier to analyse the data. Utilising the HHO approach in conjunction with the CCV fitness functions allows for the accurate assessment of the bandwidth that should be used. The RKDE method, which incorporates the Hampel function into its calculations, is used in the process of estimating the density. In the context of a traditional implementation of the Naive Bayes classifier, the training data do not change at any point throughout the process of learning. Table 5 presents the findings in their entirety.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Method** | **Precision** | **F-1** | **Recall** | **Accuracy** |
| NB | 0.654 | 0.398 | 0.286 | 0.6 |
| Robust NB | 0.831 | 0.504 | 0.362 | 0.671 |

**Table 5:** Comparison between Naive Bayes and Robust Naïve Bayes (with RKDE)

The provided results indicate a notable improvement in classification accuracy when employing the Robust Naive Bayes model in comparison to the standard Naive Bayes model. The improvement observed can be attributed to the utilization of the RKDE technique, which effectively mitigates the influence of outliers and offers a more accurate representation of data distributions. The resilience of RKDE enhances the classifier's ability to effectively handle data that is noisy and skewed, hence leading to improved outcomes in classification.

Moreover, the utilization of HHO for bandwidth selection in RKDE enhances the model's resilience to variations in the distribution of data. The selection of bandwidth values is crucial in optimizing the kernel density estimation technique, leading to enhanced accuracy in estimating the density for each class. This enhances the discriminatory capacity and classification results, especially in scenarios involving intricate and diverse data distributions.

The superior performance of the Robust Naive Bayes model may be attributed to its capacity to effectively handle outliers. The proposed solution effectively mitigates the impact of outliers on the estimation of class densities by the integration of robust M-estimation techniques with RKDE. The robustness of the system is further improved by the utilization of HHO’s adaptive bandwidth selection technique, which customizes the bandwidth based on the unique characteristics of each class's distribution. Consequently, the model exhibits increased robustness in the face of outliers, thereby enhancing its ability to accurately classify data and maintain consistency.

The experimental findings demonstrate that our proposed methodology exhibits superior performance compared to established methods in terms of both classification accuracy and resilience to outliers. The utilization of RKDE and HHO for bandwidth selection enhances the performance of the Robust Naive Bayes classifier, resulting in precise and reliable predictions. Consequently, this classifier proves to be a valuable tool for various classification tasks, particularly when dealing with challenging data distributions.

# Conclusion and Future Work

# Summary of findings

The research that was carried out led to significant new discoveries and increased comprehension in the field of accurate kernel density estimation and its implementation into the Naive Bayes classification framework. The most important findings from our analysis may be summarised in a few words as follows:

* **Optimized Robust KDE Performance:** According to the findings of the research, using the Optimised Robust Kernel Density Estimation approach results in a considerable improvement in the Naive Bayes classifier's accuracy as well as its robustness. The adaptability of the topic to a wide variety of datasets, including those that are impacted by outliers and noisy data, has been shown by means of extensive testing that has been carried out.
* **Comparison with Traditional Naïve Bayes:** When it comes to dealing with complicated datasets, the current research has produced evidence to demonstrate the superiority of the RKDE-Naive Bayes classifier in comparison to the ordinary Naive Bayes classifier. The current comparative analysis demonstrates the potential usefulness of our approach in a variety of real-world settings that are both realistic and practical.

# Contributions to the field

The fields of machine learning and data analysis both stand to benefit significantly from the multiple advancements shown in this work:

* **Enhanced Classification Performance:** Within the confines of the Naive Bayes framework, this research led to the development of a novel method that has as its primary objective the optimisation of robust kernel density estimation and bandwidth selection. When dealing with outliers and noisy data, there is an urgent need for classification algorithms that can be trusted, and this article examines that requirement.
* **Integration of Advanced Optimization Algorithms:** Within the confines of the Naive Bayes framework, this research led to the development of a novel method that has as its primary objective the optimisation of robust kernel density estimation and bandwidth selection. When dealing with outliers and noisy data, there is an urgent need for classification algorithms that can be trusted, and this article examines that requirement.
* **Practical Insights:** The RKDE approach is used for classification purposes, and our work provides useful insights into the actual implementation of this technique. This involves making suggestions for the selection of appropriate bandwidths and having an understanding of the inherent trade-offs that must be made between accuracy and the speed at which computations may be performed.

# Limitation and Future direction

Although our research has made significant strides forward, it is essential to acknowledge the inherent limits of such study:

* **Computational Overhead:** The employment of complex optimisation techniques, regardless of how effective they are, may place a significant load on the computer system. In further research initiatives, the development of the computational efficiency of our suggested approach should be prioritised as a top research priority.
* **Sensitivity to Hyperparameters:** Our methodology, much like that of other machine learning methodologies, is contingent on the hyperparameters that are selected. In further research, it will be necessary to do more research to investigate automated hyperparameter tweaking techniques.
* **Data Diversity:** A wide variety of data sets was used in the examinations that were carried out as part of our research. Nonetheless, it is vital to carry out more study in order to examine the degree to which our method may be extended to other fields in order to determine how well it works.
* **Exploration of Other Optimization Techniques:** In spite of the fact that HHO has shown its use in bandwidth selection, doing more study on various optimisation approaches can give beneficial insights into the process of reaching ideal RKDE bandwidth parameters.
* **Integration with Other Classification Algorithms:** If the recommended technique were expanded to incorporate classification algorithms other than Naive Bayes, a comparative assessment of its effectiveness in relation to the performance of other models might be carried out.

**Future Directions:**

* **Real-time Applications:** An intriguing possibility is in the investigation of ways in which our research might be incorporated into situations including real-time applications and streaming data. It is of the highest necessity to work towards the creation of algorithms that have the capacity to modify their behavior in response to shifting data distributions.
* **Interpretability:** It is of the utmost significance that our classification system be made more understandable, in particular in fields in where the openness of decision-making processes is of the highest importance, such as the medical and financial sectors.
* **Ensemble Methods:** There is a possibility that the total classification performance might be improved by investigating the possibility of merging our RKDE-Naive Bayes classifier with many other machine learning methodologies, such as ensemble methods.

In short, the purpose of this research was to devise an innovative approach to the problem of dealing with the challenges presented by noisy and outlier-filled datasets within the domain of machine learning. Our contribution consists of improving classification tools by maximising the accuracy of robust kernel density estimation and choosing the appropriate bandwidth. Even though the findings of our study mark a significant step forward, there are still several opportunities for more research and development in this area.

# References

[1] Weglarczyk, S. (2018). Kernel density estimation and its application. ITM Web of Conferences, 23, 00037.  
[2] Chevallier, E., Kalunga, E., & Angulo, J. (2017). Kernel Density Estimation on Spaces of Gaussian Distributions and  
Symmetric Positive Definite Matrices. SIAM Journal on Imaging Sciences, 10(1), 191-215.  
[3] Zeliang Zhang (2020). Big data analysis with artificial intelligence technology based on machine learning algorithm. Jour-  
nal of Intelligent and Fuzzy Systems.  
[4] Ruchi S, Pravin S (2018). Business Intelligence using Machine Learning and Data Mining techniques - An analysis. 2018  
Second International Conference on Electronics, Communication and Aerospace Technology (ICECA),.  
[5] Igor Kotenko,Igor S & Alexander B(2018). Business Intelligence using Machine Learning and Data Mining techniques -  
An analysis. Machine Learning and Big Data Processing for Cybersecurity Data Analysis,.  
[6] Domingos, P., & Pazzani, M. (1996). On the optimality of the simple Bayesian classifier under zero-one loss. Machine  
Learning, 29(2-3), 103-130.  
[7] Pérez, A., Larrañaga, P., & Inza, I. (2009). Bayesian classifiers based on kernel density estimation: Flexible classifier.  
International Journal of Approximate Reasoning, 50, 341-362.  
[8] Scott, D. W. (2015). Multivariate density estimation: theory, practice, and visualization (2nd ed.). John Wiley & Sons.  
[9] Wand, M. P., & Jones, M. C. (1995). Kernel smoothing (Vol. 60). CRC press.  
[10] Kim, J., & Scott, C. (2008). Robust Kernel Density Estimation. In Proceedings of the 22nd International Conference on  
Neural Information Processing Systems (NIPS) (pp. 3381-3384).  
[11] Maronna, R. A., Martin, D., & Yohai, V. J. (2006). Robust Statistics: Theory and Methods. John Wiley & Sons.  
[12] Huber, P. J. (1981). Robust Statistics. John Wiley & Sons.

14 N. Surname1 et al. / Optimised Robust Kernel Density Estimation for Naive Bayes Classification with Outliers  
[13] Rousseeuw, P. J., & Leroy, A. M. (2005). Robust Regression and Outlier Detection. John Wiley & Sons.  
[14] Huber, P. J., & Ronchetti, E. M. (2009). Robust Statistics (2nd ed.). John Wiley & Sons.  
[15] Hampel, F. R., Ronchetti, E. M., Rousseeuw, P. J., & Stahel, W. A. (1986). Robust Statistics: The Approach Based on  
Influence Functions. John Wiley & Sons.  
[16] Gallego, J. A., Osorio, J. F., & Gonzalez, F. A. (2022). Fast Kernel Density Estimation with Density Matrices and Random  
Fourier Features. MindLab.  
[17] Liu, B., Tan, P. N., & Zhou, J. (2022). Unsupervised Anomaly Detection by Robust Density Estimation. In Proceedings  
of the 36th AAAI Conference on Artificial Intelligence (AAAI-22) (pp. 4101-4108).  
[18] Humbert, P., Le Bars, B., & Minvielle, L. (2022). Robust kernel density estimation with median-of-means principle. In  
International Conference on Machine Learning (pp. 9444-9465). PMLR.  
[19] Vandermeulen, R. A., & Scott, C. (2014). Robust kernel density estimation by scaling and projection in Hilbert space. In  
Advances in Neural Information Processing Systems (NIPS) (pp. 3476-3484).  
[20] Calonico, S., Cattaneo, M. D., & Farrell, M. H. (2019). nprobust: Nonparametric kernel-based estimation and robust  
bias-corrected inference. arXiv preprint arXiv:1906.00198.  
[21] Wang, D., & Li, M. (2017). Robust Stochastic Configuration Networks with Kernel Density Estimation. arXiv preprint  
arXiv:1702.04459.  
[22] Wu, X. (2019). Robust likelihood cross-validation for kernel density estimation. Journal of Business & Economic Statis-  
tics, 37(4), 761-770. ———–  
[23] Wang, H., Mirota, D., & Hager, G. D. (2009). A generalized kernel consensus-based robust estimator. IEEE Transactions  
on Pattern Analysis and Machine Intelligence, 32(1), 178-184.  
[24] Boedihardjo, A. P., Lu, C. T., & Chen, F. (2015). Fast adaptive kernel density estimator for data streams. Knowledge and  
Information Systems, 42, 285-317.  
[25] Chen, Z., Fang, Z., Sheng, V., Zhao, J., Fan, W., Edwards, A., & Zhang, K. (2021). Adaptive robust local online density  
estimation for streaming data. International Journal of Machine Learning and Cybernetics, 12, 1803-1824.  
[26] Seyedali Mirjalili, Andrew Lewis “The Whale Optimization Algorithm” Advances in Engineering Software 95 (2016)  
51–67.  
[27] Zhang, W., Zhang, Z., Chao, H. C., & Tseng, F. H. (2018). Kernel mixture model for probability density estimation in  
Bayesian classifiers. Data Mining and Knowledge Discovery, 32, 675-707.  
[28] Kim, J., & Scott, C. (2009). L2 Kernel Classification. IEEE Transactions on Pattern Analysis and Machine Intelligence,  
32(10), 1822-1831.  
[29] Majdara, A., & Nooshabadi, S. (2019). Online density estimation over high-dimensional stationary and non-stationary  
data streams. Data & Knowledge Engineering, 123, 101718.  
[30] Chen J.-Y. Research on window parameter determination method based on particle swarm optimisation algorithm. (Doc-  
toral dissertation, Hebei University).  
[31] Ye, S. L., Chen, X., Xiong, Xiong Xiong, Kan, Y. J., & Wang, Z. P.. (2020). Analysis of surface air temperature in  
Jiangsu Province based on particle swarm improved adaptive kernel density estimation algorithm.Science, Technology &  
Engineering , 20(36), 8.  
[32] Hayyolalam, V., & Kazem, A. A. P. (2020). Black widow optimization algorithm: a novel meta-heuristic approach for  
solving engineering optimization problems. Engineering Applications of Artificial Intelligence, 87, 103249.  
[33] Feng, W. T., & Deng, B.. (2021). Study on global convergence analysis and parameter selection of whale optimisation  
algorithm. Control Theory and Applications.  
[34] Xu, D. G., Wang, Z. Q., Guo, Y. X., & Xing, K. J.. (2023). A review of whale optimisation algorithms. Computer Appli-  
cations Research, 40(2), 328-336  
[35] Jia L., Liu X. N., He M., & Song H. C.. (2022). A review of intelligent optimisation algorithms for quantum swarms.  
Computer Engineering and Applications, 58(7), 31-42.  
[36] Xu, P.. Improvement of the Harris Hawk optimisation algorithm and its applications. (Doctoral dissertation, Xi’an Uni-  
versity of Technology).  
[37] Zhan, Guanghan Wang, Yuhong Liu, Hao. (2022). Improved Harris Hawk optimisation algorithm with mixed strategies  
and its applications. Journal of Sensing Technology, 35(10), 1394-1403.

# Appendices

# List of papers published by the author during his degree study

# Python code display

# Created by "Boli" at 14:51, 17/08/2023 ----------%

# Email: biadboze@gmail.com %

# Github: https://github.com/biad-iritie %

# --------------------------------------------------%

from sklearn.base import BaseEstimator, ClassifierMixin

import numpy as np

from libs import kde\_lib

from libs.exp\_lib import Density\_model

from sklearn.naive\_bayes import GaussianNB

from scipy.stats import norm

list\_selection = ["hho"]

class RobustNaiveBayes(BaseEstimator, ClassifierMixin):

def \_\_init\_\_(self, h\_selection="hho") -> None:

self.class\_priors = None

self.classes = None

self.kernel = 'gaussian'

self.classifiers = {} # Store GaussianNB classifiers for each class

self.robust\_densities = {} # Store robust densities for each class

if h\_selection in list\_selection:

self.h\_selection = h\_selection

else:

raise ValueError(

'Should choose a bandwith selection between this list{}'.format(list\_selection))

def fit(self, X, y):

"""

Fit the robust Naive Bayes model with RKDE densities.

Parameters:

X (array-like): Training data features.

y (array-like): Training data labels.

"""

n\_samples, n\_features = X.shape

self.classes = np.unique(y)

n\_classes = len(self.classes)

# Calculate class priors

self.class\_priors = np.array([np.mean(y == c) for c in self.classes])

for class\_label in self.classes:

# GET for each class

class\_indices = np.where(y == class\_label)[0]

class\_data = X[class\_indices]

self.classifiers[class\_label] = GaussianNB()

self.classifiers[class\_label].fit(class\_data, y[class\_indices])

model = Density\_model("rkde", "", 0, self.kernel, kde\_lib.selection\_bandwidth(

self.h\_selection, class\_data))

model.fit(class\_data, class\_data, grid=None)

rkde = model.density

self.robust\_densities[class\_label] = rkde[:, 0]

def predict(self, X):

"""

Predict class labels and RKDE likelihoods for input data.

Parameters:

X (array-like): Input data features.

Returns:

y\_pred (array-like): Predicted class labels.

rkde\_likelihoods (array-like): RKDE likelihoods for each class.

"""

predictions = []

for sample in X:

likelihoods = []

for class\_label in self.classes:

classifier = self.classifiers[class\_label]

robust\_density = self.robust\_densities[class\_label]

# Calculate the likelihood using the robust density and GaussianNB classifier

log\_likelihood = classifier.predict\_joint\_log\_proba(

sample.reshape(1, -1))

likelihood = np.prod(

np.exp(log\_likelihood \* robust\_density.reshape(-1, 1)))

likelihoods.append(likelihood)

# Normalize likelihoods using class priors

normalized\_likelihoods = likelihoods \* self.class\_priors

predicted\_class = np.argmax(normalized\_likelihoods)

predictions.append(self.classes[predicted\_class])

return np.array(predictions)

def rkde(X\_data, X\_plot, h, type\_rho='hampel', return\_model=False):

# kernel matrix

n\_samples, d = X\_data.shape

gamma = 1. / (2 \* (h\*\*2))

Km = rbf\_kernel(X\_data, X\_data, gamma=gamma) \* \

(2 \* np.pi \* h\*\*2)\*\*(-d / 2.)

# find a, b, c via iterative reweighted least square

a = b = c = 0

alpha = 10e-8

max\_it = 100

#  first it. reweighted least ssquare with rho = absolute function

w, norm, losses = irls(Km, 'abs', n\_samples, a, b, c, alpha, max\_it)

a = np.median(norm)

b = np.percentile(norm, 75)

c = np.percentile(norm, 95)

# find weights via second iterative reweighted least square with input rho

w, norm, losses = irls(Km, type\_rho, n\_samples, a, b, c, alpha, max\_it)

#  kernel evaluated on plot data

gamma = 1. / (2 \* (h\*\*2))

K\_plot = rbf\_kernel(X\_plot, X\_data, gamma=gamma) \* \

(2 \* np.pi \* h\*\*2)\*\*(-d / 2.)

#  final density

z = np.dot(K\_plot, w)

if return\_model:

return z, w

else:

return z

def spkde(X\_data, X\_plot, h, outliers\_fraction, return\_model=False):

d = X\_data.shape[1]

beta = 1. / (1 - outliers\_fraction)

gamma = 1. / (2 \* (h\*\*2))

G = rbf\_kernel(X\_data, X\_data, gamma=gamma) \* (2 \* np.pi \* h\*\*2)\*\*(-d / 2.)

P = matrix(G)

q = matrix(-beta / X\_data.shape[0] \* np.sum(G, axis=0))

G = matrix(-np.identity(X\_data.shape[0]))

h\_solver = matrix(np.zeros(X\_data.shape[0]))

A = matrix(np.ones((1, X\_data.shape[0])))

b = matrix(1.0)

sol = solvers.qp(P, q, G, h\_solver, A, b)

a = np.array(sol['x']).reshape((-1, ))

# final density

GG = rbf\_kernel(X\_data, X\_plot, gamma=gamma) \* \

(2 \* np.pi \* h\*\*2)\*\*(-d / 2.)

z = np.zeros((X\_plot.shape[0]))

for j in range(X\_plot.shape[0]):

for i in range(len(a)):

z[j] += a[i] \* GG[i, j]

if return\_model:

return z, a

else:

return z

class HHO\_BandwidthSelection(Problem):

def \_\_init\_\_(self, lb, ub, minmax, \*\*kwargs):

super().\_\_init\_\_(lb, ub, minmax, \*\*kwargs)

self.lb = lb

self.ub = ub

self.minmax = minmax

self.data = kwargs["data"]

def obj\_func1(self, bandwidth):

return bcv\_objective(bandwidth, self.data)

def obj\_func2(self, bandwidth):

return ls\_ucv\_objective(bandwidth, self.data)

def fit\_func(self, solution):

return [self.obj\_func1(solution), self.obj\_func2(solution)]

def get\_name(self):

return "HHO for Bandwidth Selection"

def hho\_bandwith\_selection(data):

problem\_multi = HHO\_BandwidthSelection(

lb=np.array([.1]), ub=np.array([1]), minmax="min", obj\_weights=[1, 1], data=data)

# Define the model and solve the problem

# epoch = 1000

epoch = 10 # 50 maximum number of iterations

# pop\_size = 50

pop\_size = 10 # 10 number of population size

model = OriginalHHO(epoch, pop\_size)

best\_position, best\_fitness = model.solve(problem\_multi)

print("Result hho\_bandwith\_selection: {}".format(best\_position))

return best\_position[0]