Computationally Lightweight Classifiers with Frequentist Bounds on Predictions

Shreeram Murali
Cyber-physical Systems Group
Aalto University

Cristian R. Rojas
Decision and Control Systems
KTH Royal Institute of Technology

Dominik Baumann
Cyber-physical Systems Group
Aalto University

Abstract

While both classical and neural network classifiers can achieve high accuracy, they fall short on offering uncertainty bounds on their predictions, making them unfit for safety-critical applications. Existing kernel-based classifiers that provide such bounds scale with $\mathcal{O}(n^{\sim 3})$ in time, making them computationally intractable for large datasets. To address this, we propose a novel, computationally efficient classification algorithm based on the Nadaraya-Watson estimator, for whose estimates we derive frequentist uncertainty intervals. We evaluate our classifier on synthetically generated data and on electrocardiographic heartbeat signals from the MIT-BIH Arrhythmia database. We show that the method achieves competitive accuracy >96% at $\mathcal{O}(n)$ and $\mathcal{O}(\log n)$ operations, while providing actionable uncertainty bounds. These bounds can, e.g., aid in flagging low-confidence predictions, making them suitable for real-time settings with resource constraints, such as diagnostic monitoring or implantable devices.

1 INTRODUCTION

Supervised classification is the basis for various categorization and instance-counting problems. Several classes of methods exist that combine high accuracy and computational efficiency. Approaches in classical machine learning, such as Logistic Regression or Support Vector Machines (SVMs) (Shalev-Shwartz and Ben-David, 2014; Schölkopf and Smola, 2001), offer high accuracy with low computational cost. Nevertheless, deep learning methods are the de facto state-of-the-art for high accuracy; however, these models are often 'black boxes', providing point predictions without an explainable confidence measure (Longo et al., 2024). To be adopted in data-abundant and safety-critical applications, classifiers must jointly address (i) accuracy, (ii) computational efficiency, and (iii) reliability.

In classical machine learning, most methods lack reliable uncertainty quantification, preventing their adoption in high-stakes environments. Some classifiers, for example, SVMs, draw boundaries through data to delineate classes (Shalev-Shwartz and Ben-David, 2014; Schölkopf and Smola, 2001). However, this is insufficient for uncertainty quantification; for instance, the classifier might be highly uncertain about a prediction near a boundary. Soft classifiers assign probabilities to each class (Shalev-Shwartz and Ben-David, 2014; Rasmussen and Williams, 2008; Baumann and Schön, 2024). Although the latter conveys more information about certainty, the probabilities cannot be interpreted as true confidence levels *sua sponte* as they are often poorly calibrated (e.g., a model can be highly inaccurate while reporting high confidence). To address this, various uncertainty quantification techniques such as bootstrapping or deep ensembles have been employed, which train multiple models to estimate predictive variance (Hall, 1988; Gal and Ghahramani, 2016). While useful, these methods are often heuristic, computationally expensive, and typically do not provide formal guarantees on the prediction error.

For a more theoretically motivated approach to uncertainty quantification, Bayesian non-parametric methods, most notably Gaussian Process (GP) classification, are used (Rasmussen and Williams, 2008). GPs provide a full posterior distribution over the class probabilities; however, exact GP inference is intractable in the classification setting, and even the approximate methods scale as $\mathcal{O}(n^3)$ with the number of data points. Furthermore, there are other disadvantages as-

Proceedings of the 29th International Conference on Artificial Intelligence and Statistics (AISTATS) 2026, Tangier, Morocco. PMLR: Volume 300. Copyright 2026 by the author(s).

sociated with a Bayesian framework beyond computational intractability; namely, a Bayesian model reflects its updated beliefs over a prior rather than a repeatable frequentist error interval.

To address the latter, Baumann and Schön (2024) derive frequentist uncertainty intervals using conditional kernel mean embeddings in a classification task. These embeddings are, however, computationally similar to GPs in that they are inefficient due to the need for matrix inversion; thus, the classifier from Baumann and Schön (2024) comes at the same cost as GP regression.

Contributions. To address the joint challenge of computational efficiency and rigorous frequentist uncertainty intervals, we propose a classifier that uses the Nadaraya-Watson (NW) estimator (Nadaraya, 1964; Watson, 1964). The NW estimator is a kernel density estimator whose typical use case is regression. In this paper, we reformulate the estimator as a classifier. We then derive frequentist bounds on its prediction errors and provide computationally lightweight implementations that further enhance its superior linear complexity. Our main contributions are:

- A non-parametric classification algorithm based on the Nadaraya-Watson estimator that scales linearly, $\mathcal{O}(n)$, with the size of the training set.
- The derivation of frequentist uncertainty bounds on the estimated class probabilities, valid for both overlapping data distributions and well-separated distributions.
- Computationally efficient variations on the naive implementation to improve from linear to sublinear and logarithmic computational complexity.
- Validation on both synthetic and real-world medical data, demonstrating that our method achieves competitive accuracy while providing actionable uncertainty bounds at a fraction of the computational cost of competing methods.

2 PROBLEM SETTING

We define the underlying probability space as $(\mathbf{\Omega}, \mathcal{F}, \mathcal{P})$ with random variables $Y : \mathbf{\Omega} \to \mathcal{Y}$ and labels $C : \mathbf{\Omega} \to \mathcal{C}$ that take values in $\mathcal{Y}$ and $\mathcal{C}$ respectively. Here, $\mathbf{\Omega}$ is the sample-space, $\mathcal{F}$ is a sigma-algebra on $\mathbf{\Omega}$, and $\mathcal{P}$ is a measure that assigns probabilities to events in $\mathcal{F}$.

In this work, we estimate $p_c(y) \coloneqq \mathbb{P}(C = c \mid Y = y)$, the probability of observations $y \in \mathcal{Y} \subseteq \mathbb{R}^d$ belonging to class $c \in C \in \mathbb{N}$. We also derive high probability uncertainty bounds on the estimate of p_c. That is, for each class c and a user-defined probability of at least $1 - \delta$, we show that the error between the true probability p_c and its estimate $\hat{p}_c$ is bounded as a function of y, δ, and sample size n, as

$$|p_c(y) - \hat{p}_c(y)| \leq \epsilon_c(y, \delta, n). \tag{1}$$

Since the nature of the true probability function $p_c(y)$ is not known, we introduce assumptions about the underlying data distribution from which observations y have been sampled. These assumptions cover two cases: one where the underlying data distribution is overlapping, and the other where it is separable. Here, we note that only one of the following two assumptions about the distribution must hold.

Overlapping distributions. For overlapping distributions, we assume the underlying probability function is Lipschitz continuous. This assumption captures scenarios wherein different classes may share similar characteristics yet remain distinguishable through smooth transitions in the probability space.

Assumption 1. *The true probability function $p_c(y)$ is Lipschitz continuous with a known Lipschitz constant $L < \infty$. That is, for each $c \in \mathcal{C}$, and any two samples y and y',*

$$|p_c(y) - p_c(y')| \leq L\|y - y'\|. \tag{2}$$

In this paper, we use $\|\cdot\|$ to denote the L^2 norm. Thus, $\|y - y'\|$ represents the Euclidean distance between the two samples y and y'.

Remark 1. *Assuming knowledge of the Lipschitz constant a priori is common in the control and safe learning literature (Magureanu et al., 2014; Brunke et al., 2022). It can generally be approximated from data, to which end the majority of existing approaches use the classical estimator from Strongin (1973):*

$$\hat{L} \coloneqq r \max_{i \neq j} \frac{|f(x_i) - f(x_j)|}{\|x_i - x_j\|}, \tag{3}$$

where $r \in \mathbb{R}$ is a multiplicative factor, $(x_i, f(x_i))$ is a data sample, and f is the unknown function to be estimated.

In our experiments, we use an approach similar to (3) (see Appendix D) to approximate a Lipschitz constant. Nevertheless, various approaches have been built on Strongin (1973); for instance, Wood and Zhang (1996) fit an approximate distribution to the Lipschitz estimate in the one-dimensional case, and Sergeyev (1995) uses this approach to the multi-dimensional case by using space-filling curves to solve a global optimization problem. Further, Novara et al. (2013) and Calliess et al. (2020) extend this estimator to handle bounded

observational noise, while Huang et al. (2023) provide finite-sample guarantees on the estimate with stronger assumptions on the target function.

Yet, the common theme in these approaches is to invoke a regularity assumption on the underlying unknown function. This is what we do through Assumption 1.

Separable distributions. For separable distributions, we assume that samples from different classes are well-separated in the feature space, with a known minimum distance between them.

Assumption 2. *Samples in $\mathbf{\Omega}$ are separable with a known margin γ. That is, for any two samples (y, c) and (y', c') drawn from $\mathcal{D}$, where $c \neq c'$,*

$$\|y - y'\| \geq \gamma, \tag{4}$$

with probability 1.

In other words, the distribution $\mathcal{D}$ has a margin γ if the distance between points with differing labels is at least γ.

Nature of measurements. Next, we require an assumption about the nature of sampling from our dataset.

Assumption 3. *Samples (y_i, c_i) are independently drawn from the same distribution $\mathcal{D}$.*

This independent and identical distribution (i.i.d.) assumption is often invoked in the literature to provide theoretical guarantees (Rao and Protopopescu, 1996). In our case, we introduce this assumption to derive data-dependent bounds on the sampling error.

3 CLASSIFIERS

In this section, we first formulate the NW estimator as a classifier in Section 3.1, derive bounds on the estimate in Section 3.2, and propose improvements to an already efficient naive implementation in Section 3.3.

3.1 Nadaraya-Watson classifier

From the standard form of the Nadaraya-Watson estimator (Nadaraya, 1964; Watson, 1964), we replace the real-valued observation with the indicator function $\mathbb{1}_{c_i} := \mathbb{1}\{c_i = c\}$, a one-hot vectorized representation of the class label c_i. In doing so, we reformulate the Nadaraya-Watson estimator to estimate the probabilities $\mathbf{p}_c(y)$ as

$$\hat{\mathbf{p}}_c(y) = \frac{1}{\kappa_n(y)} \sum_{i=1}^{n} K_\lambda(y, y_i) \mathbb{1}_{c_i}, \tag{5}$$

where

$$\kappa_n(y) := \sum_{i=1}^{n} K_\lambda(y, y_i),$$
$$K_\lambda(y, y_i) := \frac{1}{c_k} K\left(\frac{\|y - y_i\|}{\lambda}\right). \tag{6}$$

Here, $K(\cdot)$ is the kernel function that conveys the degree of similarity between a query sample y and training sample y_i, λ is the user-defined bandwidth parameter, c_k is a normalization constant, and $\mathbb{1}_{c_i} := \mathbb{1}\{c_i = c\}$ is the one-hot vectorized representation of the class label c_i. We use $\mathbf{p}_c(y)$ to denote the vector of assigned class probabilities across all classes, and $p_c(y)$ to describe the scalar probability for a single class c.

Assumption 4. *The kernel $K : \mathbb{R}^d \to \mathbb{R}$ is non-negative and bounded such that for any $v \in \mathbb{R}^d$ and some $c_k < \infty$, $0 \leq K(v) \leq c_k$, and $K(v) = 0$ for all $\|v\| > 1$.*

Since the kernel function is user-defined, we can satisfy this assumption by explicitly selecting a suitable kernel or formulating one in line with this assumption. For instance, many popular kernels, such as the boxcar or the Epanechnikov kernel, already meet this assumption; for those that do not, such as the Gaussian (Radial Basis Function; RBF) kernel, the outputs for $\|v\| > 1$ can be explicitly truncated.

Furthermore, in Appendix A.3 we relax Assumption 4 to include non-bounded kernels in exchange for a new assumption on bounded inputs to extend the validity of our bounds to kernels with infinite support.

The computational time of the NW estimator scales linearly with a naive implementation, but as we show in Section 3.3, this can be improved to sublinear computational complexity with some preprocessing.

3.2 Deriving bounds on the estimates

We provide theoretical worst-case bounds on the estimate produced in (5) by splitting the error into two sources: the uncertainty that is inherent in not obtaining samples of the true probability function $p_c(y)$ but only discrete labels, and the sampling error that arises when estimating a function from a finite number of samples. We refer to the former as the classifier's *bias* and the latter as its *sampling error*.

In this section, we derive bounds on the error between the true probability function $p_c(y)$ and its estimate $\hat{p}_c(y)$ as depicted in (1). We start by deriving the estimator's bias corresponding to the two cases in Assumptions 1 and 2.

To this end, we introduce a virtual estimate $\bar{p}_c(y)$, a quantity that could hypothetically be determined if we

had true probability measurements instead of discrete labels. With this, we split (1) into two terms, one corresponding to the estimator's bias and the other to its statistical error in sampling:

$$|p_c(y) - \hat{p}_c(y)| \leq \underbrace{|p_c(y) - \bar{p}_c(y)|}_{\text{bias}} + \underbrace{|\bar{p}_c(y) - \hat{p}_c(y)|}_{\text{sampling error}}. \tag{7}$$

We prove that these two terms are individually bounded. Then, by the triangle inequality, the term on the left side of (7) would also be bounded.

3.2.1 Bias

We first analyse the bias term under the two scenarios mentioned in Section 2: when the underlying probability function is Lipschitz continuous, and when the data distributions are separable.

Lemma 1. *Under Assumptions 1 and 4, we have, for all $n \geq 0$ and $y \in \mathcal{Y}$,*

$$|p_c(y) - \bar{p}_c(y)| \leq L\lambda, \tag{8}$$

where L is the known Lipschitz constant from (2) and λ is the user-defined kernel bandwidth from (6).

The proofs of Lemma 1 and further technical results are collected in Appendix A and B. This lemma shows that for overlapping distributions, the bias of our estimator is bounded by the product of the Lipschitz constant and the kernel bandwidth. Intuitively, this means that the bias increases linearly with both the rate of change of the probability function and the size of the local neighbourhood we consider for estimation.

Lemma 2. *Under Assumptions 2 and 4, we have for almost every $y \in \mathcal{Y}$*

$$|p_c(y) - \bar{p}_c(y)| \leq \frac{\lambda}{\gamma}, \tag{9}$$

where λ is the user-defined kernel bandwidth parameter as depicted in (6) and γ, in accordance with Assumption 2, is the known margin of the distribution $\mathcal{D}$.

For separable distributions, (9) demonstrates that the bias is bounded by the ratio of the kernel bandwidth to the margin between classes. This suggests that larger margins between classes allow for larger kernel bandwidths while maintaining the same bias bound.

Additionally, this result can be extended to positive-definite kernels with infinite support. To do this, we trade the compactness of the kernel in Assumption 4 for an assumption on bounded input space.

Assumption 5. *There exists a finite diameter Φ such that for any sample y_i and input $y \in \mathcal{Y}$,*

$$\|y - y_i\| \leq \Phi. \tag{10}$$

This assumption lets us bound the bias of the classifier with an extra term that represents the kernel-weighted sum of the number of samples outside a chosen bandwidth λ^*.

Lemma 3. *Under Assumptions 1, 2, 4 with the change that $K(v) \geq 0$ for all $\|v\| > 1$, and 5 we have for almost every $y \in \mathcal{Y}$*

$$|p_c(y) - \bar{p}_c(y)| \leq \beta\lambda^* + \beta\Phi\varepsilon_t, \tag{11}$$

where

$$\varepsilon_t := \sum_{i \in \mathcal{I}_{\text{far}}} \frac{K_\lambda(y, y_i)}{\kappa_n(y)} \|y - y_i\|$$

is a term that represents the weighted sum corresponding to the samples in the tail of the kernel's span (indices $\mathcal{I}_{\text{far}}$, where $\|y - y_i\| > \lambda^*$), Φ is the input space diameter from Assumption 5, and $\beta = L$ or $1/\gamma$ depending on whether we assume an overlapping (see Assumption 1) or a separable (see Assumption 2) distribution on $\mathcal{D}$.

We prove this Lemma in Appendix Section A.3. However, since this is a more conservative bound, we use the bounds from Lemmas 1 and 2 in our experiments (Section 4). Since there is greater freedom in choosing a kernel than in choosing data, we consider Lemma 3 as a theoretically valid extension applicable to a few limited datasets.

3.2.2 Sampling error

Having bounded the bias term, we now analyse the sampling error, which captures the uncertainty due to finite sample estimation.

Lemma 4. *Under Assumptions 3 and 4, we have, for all $n \geq 0$, with probability at least $1 - \delta$,*

$$|\bar{p}_c(y) - \hat{p}_c(y)| \leq 2\sigma \frac{\alpha_n(y, \delta)}{\kappa_n(y)}, \tag{12}$$

where

$$\alpha_n(y, n) = \begin{cases} \sqrt{\kappa_n(y) \log(\delta^{-1}\sqrt{1 + \kappa_n(y)})}, \\ \qquad\qquad \text{if } \kappa_n(y) > 1, \\ \sqrt{\log\left(\sqrt{2}/\delta\right)}, \\ \qquad\qquad \text{if } 0 < \kappa_n(y) \leq 1. \end{cases} \tag{13}$$

This lemma provides a probabilistic bound on the sampling error that depends on the local density of samples through $\kappa_n(y)$. The bound becomes tighter in regions with more samples (larger $\kappa_n(y)$), reflecting the intuition that predictions are more reliable in data-dense regions. The two cases in the definition of α_n handle differently the scenarios of sparse and dense sampling.

3.2.3 Combined bounds

Correspondingly, we now formulate overall bounds on the prediction error by gathering the bounds from (8), (9), and (13).

Corollary 1. *Under the same assumptions as Lemma 4 and either Lemma 1 or Lemma 2, we have, with a probability of at least* $1-\delta$,

$$|p_c(y) - \hat{p}_c(y)| \leq \beta\lambda + 2\sigma\frac{\alpha_n(y,\delta)}{\kappa_n(y)}, \tag{14}$$

where $\alpha_n(y,\delta)$ *is the data-dependent term from* (13) *and* $\beta = L$ *or* $1/\gamma$ *based on the nature of the data's underlying probability distribution.*

Proof. This follows from applying the triangle inequality and collecting the bias term from either Lemma 1 or 2, and the sampling error from Lemma 4. □

3.3 Computational efficiency improvements

The naive implementation of the proposed classifier scales with $\mathcal{O}(n)$, which is a significant improvement over the closest competing alternative by Baumann and Schön (2024), which scales with $\mathcal{O}(n^3)$. Nevertheless, we can further improve its efficiency by taking inspiration from k-nearest neighbour-based methods (Nnyaba et al., 2024) and implement a *localized* variant. To this end, we employ k-d trees to facilitate the lookup of k-nearest neighbours. Building a k-d tree takes $\mathcal{O}(n\log n)$ time, and querying scales with $\mathcal{O}(k+\log n)$. Thus, for small k, we have approximately logarithmic scaling behaviour, while for increasing k, the outputs and the performance of the *localized* classifier approach that of the *regular* implementation.

As an alternative, we implement a hash table variant of the proposed classifier in (5) that performs lookup with $\mathcal{O}(\log n)$ complexity. We refer to this implementation in successive text as the *dyadic* classifier. The construction of the hash table scales with $\mathcal{O}(n)$. However, this approach suffers from two crucial limitations. First, it does not allow for the efficient computation of bounds. This is because $\kappa_n(y)$ is a query-dependent value: it is the sum of weights calculated from the distances between a new query sample and the training samples. Range trees and hash tables are data structures built solely on training data; they cannot perform distance-based calculations relative to a new query sample y. Second, this approach suffers from the curse of dimensionality. For a user-defined resolution parameter m, the number of dyadic cells scales with increasing feature dimension d in the order of 2^{m^d}.

We compare the theoretical time complexities of the proposed classifier, its variants, and baseline methods Logistic Regression (Shalev-Shwartz and Ben-David, 2014) and CMEs (Baumann and Schön, 2024) in Table 1.

Table 1: Time complexity of different NWC implementations compared with CMEs and Logistic Regression.

	Preprocessing	Querying
Regular	–	$\mathcal{O}(n)$
Dyadic	$\mathcal{O}(n)$	$\mathcal{O}(\log n)$
Localized	$\mathcal{O}(n\log n)$	$\mathcal{O}(k+\log n)$
CME	–	$\mathcal{O}(n^3)$
LR	$\mathcal{O}(nd)$	$\mathcal{O}(d)$

Remark 2. *Rao and Protopopescu (1996) also present a variant of the NW estimator, improving from the naive* $\mathcal{O}(n)$ *to a faster* $\mathcal{O}((\log n)^d)$ *implementation using range trees. With this data structure, the method enables the computation of the number of points within an arbitrary hyper-rectangle. However, this is misaligned with the objectives of classification; the estimation of any point within a cube is a pre-calculated aggregate of the training data that fell into said cube. Thus, the prediction problem is not one of searching in an arbitrary range; it is one of a direct lookup operation. These operations are better handled using hash tables.*

Remark 3. *Since* $K(v) := 0$ *for* $v \geq 1$ *(Assumption 4), we can interpret the regular classifier as a local estimator whose locality is defined by the bandwidth parameter. The localized implementation is thus an alternative framing of the regular classifier, wherein we look up the* k*-nearest neighbours and compute their weights rather than iterating through the entire dataset to compute kernel-weights for samples within the bandwidth.*

4 EXPERIMENTS

We implement and evaluate our proposed classifier and its variants against baseline methods in two settings: synthetically generated data to validate our theoretical assumptions, and a real-world electrocardiogram (ECG) dataset to demonstrate its practical utility in a safety-critical application. Additionally, we compare the methods on the MNIST dataset in Appendix E.3. The code is available in the supplementary material.

Synthetic data. We create two datasets, one Lipschitz continuous, and another separated by a margin. For the former, we define the true underlying probability function $p_c(y)$ using a logistic function with a direct relationship to L (see Appendix C). We sample points from the probability function such that it matches Assumption 1. For the latter, we generate class centres

github.com/shreeram-murali/AISTATS-2026

spaced by a margin greater than γ and sample points from a small cluster surrounding the centre. Figure 1 shows the two datasets used in our results.

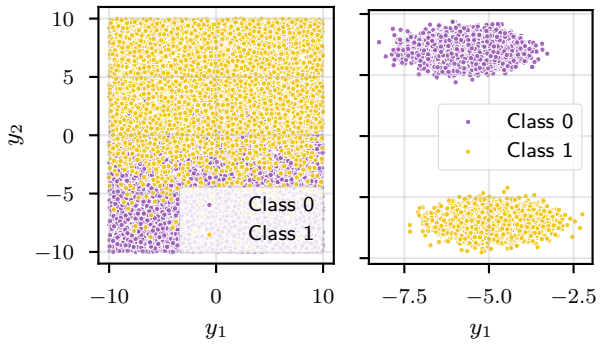


Figure 1: The two synthetic datasets used for evaluating the classifier. *One dataset with overlapping classes, adhering to Assumption 1 (left), and one with classes separated by a known margin, in accordance with Assumption 2 (right).*

Electrocardiographic data. We use the MIT-BIH Arrhythmia database, a widely-used benchmark in cardiology research (Moody and Mark, 2001). The dataset contains approximately 110 000 annotated heartbeats from 48 half-hour recordings. Each heartbeat sample is represented by a 187-dimensional vector, which we truncated to contain the first 100 features to retain the most informative part of the QRS complex. We followed data preprocessing steps identical to those in prior work (e.g., Nnyaba et al. (2024)). The dataset was split into 87 554 training and 21 892 testing samples. Heartbeats are categorized into five classes[1] as: Normal (N), Supraventricular ectopic (S), Ventricular ectopic (V), Fusion (F), and Unclassifiable (Q). The dataset exhibits a significant class imbalance, which reflects real-world clinical scenarios.

4.1 Results

For the synthetic overlapping dataset, we set $L = 0.15$, and for the separable dataset, we set the margin as an equivalent $\gamma = 6.67$. For the *regular* and *localized* NWC variants, we use the Epanechnikov kernel, defined as

$$K(v) = \begin{cases} 1 - v^2, & \text{if } \|v\| \leq 1, \\ 0, & \text{otherwise,} \end{cases} \tag{15}$$

with bandwidth $\lambda = 0.2$. In Appendix Section F we provide an ablation study to find the best kernel and λ that maximizes our classifier's accuracy and bounds.

[1]defined by the Association for the Advancement of Medical Instrumentation (AAMI) EC57 standard

The results in Figure 2 depict the performance of the proposed and baseline methods. Here, we see that CMEs' computational costs were prohibitively expensive; we run the classifier only up to a maximum of 10 000 samples. The top row of figures shows the superior computational efficiency of the *regular*, *localized*, and *dyadic* variants over CMEs. From the second row of Figure 2, we see the uncertainty intervals tightening significantly with increasing n. Additionally, for accuracy, we observe that the computational efficiency of the proposed classifiers does not come at the cost of requiring larger training sets.

Next, we summarize results from electrocardiographic data. On the MIT-BIH dataset, we set the Lipschitz constant to $L = 0.05$ and the kernel bandwidth to $\lambda = 0.75$ (see Appendix Section D) with the Epanechnikov kernel. Figure 3 shows the performance of the *regular* and *localized* classifier on this dataset, with illustrative examples of predictions, bounds, and the assigned class probabilities. When trained on the full dataset of over 87 554 samples, our *regular* classifier achieves an accuracy of 96.2%, while the *localized* one has an even higher accuracy of 97.8%. This comparison, along with precision-recall scores and runtimes, is shown in Table 2 in a detailed evaluation of the classifiers with varying training set sizes. While CMEs provide tighter bounds at smaller set sizes, their cubic complexity makes them intractable for the full dataset. The proposed classifier, even with a naive implementation, was over 500 times faster to evaluate on the entire 87 554 sample strong dataset than CMEs were on just 10 000 samples.

Crucially, the uncertainty bounds provided by the proposed classifiers are actionable. As seen in Figure 3(c), incorrect predictions are typically associated with lower confidence (a lower probability estimate $\hat{p}_c$) or higher uncertainty. Correspondingly, in Figure 3(e) we show the cumulative recall curve; for instance, we observe that flagging the top 10% of the most uncertain predictions captures roughly 40% of the classifiers' incorrect predictions. This allows for a system where uncertain predictions can be automatically flagged for manual review by a clinician, a critical feature for deployment in healthcare. The *localized* NWC emerges as the most accurate and computationally efficient method, while the *regular* NWC provides stronger theoretical bounds. Both variants significantly outperform the baselines in achieving a practical balance of accuracy, efficiency, and reliability.

Each experiment in this section addresses a specific case: the synthetic data allows for the creation of datasets with known Lipschitz-continuity and margins, and the ECG data highlights the NW classifiers' core competence in balancing accuracy, relia-

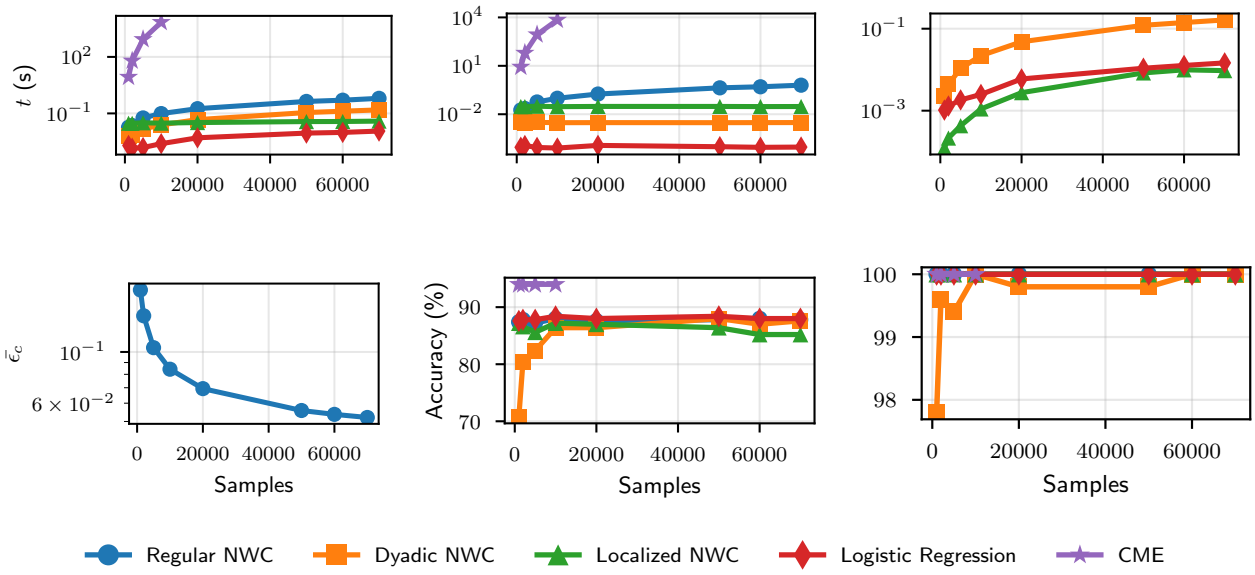
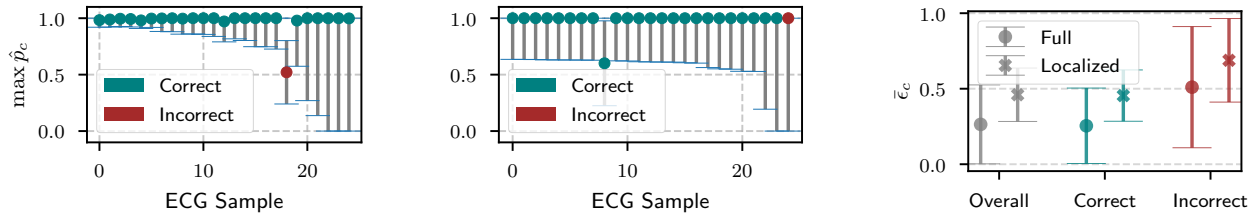


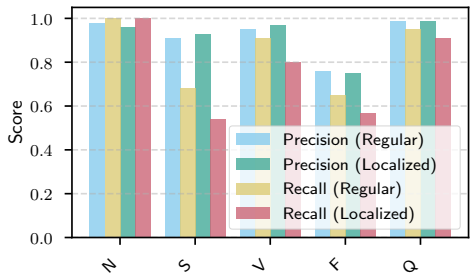
Figure 2: Performance of the proposed classifiers on the synthetic datasets compared to baselines with varying sample sizes. *We plot total runtime, prediction time, fit time (top row); average bounds $\bar{\epsilon}_c$ for $\delta = 0.05$ for all classes, accuracy on the overlapping and separable dataset (bottom row). We observe that our algorithm is significantly more sample-efficient than the CME-based classifier, while achieving high accuracy with minimal uncertainty.*



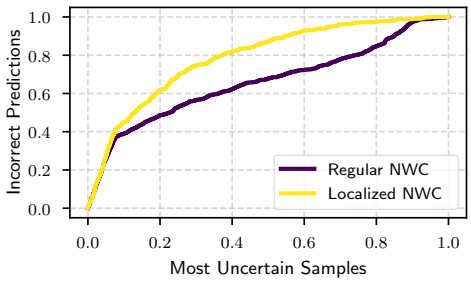
(a) *Regular* NWC: predicted probabilities and 95% bounds ($\delta = 0.05$).

(b) *Localized* NWC: predicted probabilities and 95% bounds ($\delta = 0.05$).

(c) Mean bound width vs. predicted probability.



(d) Precision-recall metrics for all classes and both proposed classifier variants: *regular* and *localized* NWC.



(e) The cumulative recall curves. *The curves show the proportion of total errors identified when samples are ranked by descending uncertainty.*

Figure 3: Mean uncertainty intervals, precision-recall metrics for the proposed classifiers, and waveforms. *Our classifier shows higher uncertainty in misclassified labels, as well as high precision and recall scores.*

Table 2: Acccuracy and runtime comparison of the classifiers on the ECG dataset. *We observe that both our approaches outperform logistic regression and the CME-based classifier in terms of accuracy, while being computationally orders of magnitude cheaper than the latter.*

Size	Classifier	Accuracy	Precision	Recall	Precompute (s)	Query Time (s)
1 000	***Regular NWC***	86.4	0.881	0.864	–	0.140
	Localized NWC	91.0	**0.903**	0.910	**0.0008**	0.085
	Logistic Regression	88.8	0.859	0.888	0.0128	**0.0002**
	CME	**92.0**	0.846	**92.0**	–	8.883
10 000	***Regular NWC***	93.4	0.938	0.934	–	1.573
	Localized NWC	**96.4**	**0.963**	**0.964**	**0.0145**	0.395
	Logistic Regression	91.6	0.894	0.916	0.161	**0.0002**
	CME	92.0	0.846	0.920	–	8618.85
60 000	***Regular NWC***	95.6	0.958	0.956	–	11.607
	Localized NWC	**97.6**	**0.977**	**0.976**	**0.188**	1.729
	Logistic Regression	91.0	0.908	0.910	5.929	**0.0003**
	CME	–	–	–	–	–
87 554	***Regular NWC***	96.2	0.965	0.962	–	16.905
	Localized NWC	**97.8**	**0.979**	**0.978**	**0.318**	1.691
	Logistic Regression	91.0	0.903	0.910	6.527	**0.0002**
	CME	–	–	–	–	–

bility, and computational efficiency. The proposed classifier successfully bridges the gap between efficient but non-guaranteed methods like Logistic Regression, and methods with formal guarantees like CMEs, which are computationally prohibitive (Baumann and Schön, 2024). Our proposed variants—*localized* and *dyadic* implementations—further offer a tunable trade-off: they provide higher computational efficiency at the cost of more conservative bounds. In regulated industries like healthcare, where diagnosticians often manually review data, it is desirable to have methods that are accurate and also express uncertainty. Here, the role of bounds becomes clearer: measurements can thus be flagged for manual review based on the strength of their bounds.

5 RELATED WORK

In this section, we contextualize our contributions and position our work against the existing literature in uncertainty quantification, non-parametric methods, and previous work on ECG heartbeat classification.

Frequentist uncertainty quantification. Many contemporary uncertainty quantification techniques, particularly in deep learning, rely on empirical methods like Monte-Carlo Dropout or deep ensembles (Gal and Ghahramani, 2016; Shahid et al., 2024). These approaches are empirical and lack the formal, high-probability error guarantees necessary for high-stakes decision-making. On the other hand, while Bayesian methods offer a more principled method for uncertainty quantification, they are non-frequentist, and are computationally and analytically intractable (Rasmussen and Williams, 2008; Villacampa-Calvo et al., 2021). The most direct frequentist predecessor to our work, the CME-based classifier by Baumann and Schön (2024), provides the desired formal guarantees but suffers from the same prohibitive $\mathcal{O}(n^3)$ complexity due to its reliance on matrix inversion. Our work directly addresses this limitation.

Non-parametric kernel regression. Our choice of the Nadaraya-Watson estimator is motivated by its history as a non-parametric regressor (Nadaraya, 1964; Watson, 1964). It has typically found its usage in statistical applications (Schuster and Yakowitz, 1979; Nadaraya, 1989; Prakasa Rao, 1983). Moreover, the estimator has also been applied to system identification problems in control theory (Schuster and Yakowitz, 1979; Juditsky et al., 1995; Ljung, 2006; Mzyk and Wachel, 2020). Nevertheless, the most relevant usage of the NW estimator to this paper's contributions has been in the safe learning literature as a computationally efficient alternative to GPs (Baumann et al., 2023). It has been successfully used to provide safety and optimality guarantees in regression settings for reinforcement learning and control (Kowalczyk et al., 2024; Baumann et al., 2025). Nevertheless, we are not aware of any previous work that reformulates the NW estimator as a multi-class classi-

fier and derives frequentist uncertainty bounds on its class probability estimates. In doing so, we inherit the computational efficiency of the estimator and couple it with the formal guarantees typically found in the domain of computationally expensive methods.

Approaches to arrhythmia detection. The implications of our work are most evident in the context of our ECG experiments. Classification in the context of arrhythmia detection spans a large class of problems: some methods aim to classify strips of rhythms (Hedén et al., 1996), comprising multiple heartbeats, while others aim to classify individual beats themselves (Kachuee et al., 2018). Many classical and deep learning approaches have been successful in yielding highly accurate predictions on the MIT-BIH dataset, ranging from 95.9% to 99.87% (Kumari and Sai, 2022; Zhou and Fang, 2024; Abdalla et al., 2020; Gao et al., 2019; Jha and Kolekar, 2020), but these approaches often disregard the uncertainty associated with predictions (see Appendix E.5). Some recent studies have attempted to quantify uncertainty through Monte Carlo dropout simulations (Zhang et al., 2022) and variational encoders (Barandas et al., 2024). The most relevant contribution by Nnyaba et al. (2024) utilized a k-nearest neighbours based GP classification method (Muyskens et al., 2021). While Nnyaba et al. (2024) demonstrate strong results on the ECG dataset, their bounds are less rigorous, non-frequentist, and the classifier scales cubically with the number of neighbours k considered.

6 CONCLUSIONS

We have presented a novel classification algorithm that reformulates the Nadaraya-Watson estimator for multi-class classification tasks. Our work jointly addresses three challenges in modern machine learning: computational efficiency, theoretical guarantees, and practical applicability in safety-critical domains. The classifier achieves linear time complexity, $\mathcal{O}(n)$, a significant improvement over the cubic scaling of existing methods that provide formal guarantees. This efficiency does not come at the cost of reliability; we derive rigorous frequentist bounds on prediction errors, providing guarantees for both overlapping and separable data distributions.

We complement the theoretical contributions with practical implementations that further enhance computational efficiency. Our localized variant, leveraging k-d trees, achieves sublinear time complexity while maintaining high accuracy, reaching 97.8% on real-world ECG data, though with more conservative bounds. The *dyadic* implementation offers even faster lookups at $\mathcal{O}(\log n)$, however, without the ability to specify kernels or compute bounds. These variants provide practitioners with flexible options to balance computational resources against uncertainty quantification needs.

Future work. Despite promising results, our approach has limitations suggesting directions for future research. A primary challenge is estimating the Lipschitz constant from data, which can be non-trivial (Wood and Zhang, 1996; Tokmak et al., 2025). Another challenge would be to overcome the expressivity of Euclidean distances in higher dimensions. Future work could also explore special cases—for instance, ordinal and sequential classification—to tighten theoretical bounds, enhancing the classifier's utility.

Acknowledgements

This research was partially supported by the Research Council of Finland flagship programme: the Finnish Center for Artificial Intelligence (FCAI), the Tandem Industry Academia Seed funding from the Finnish Research Impact Foundation, the Swedish Research Council under contract number 2023-05170, the Wallenberg AI, Autonomous Systems and Software Program (WASP), and the Swedish Civil Defence and Resilience Agency (Project MAD-VAMCHS). We also acknowledge the computational resources provided by the Aalto Science–IT project.

References

Abbasi-Yadkori, Y., Pal, D., and Szepesvari, C. (2011). Online least squares estimation with self-normalized processes: An application to bandit problems.

Abdalla, F. Y. O., Wu, L., Ullah, H., Ren, G., Noor, A., Mkindu, H., and Zhao, Y. (2020). Deep convolutional neural network application to classify the ECG arrhythmia. *Signal, Image and Video Processing*, 14(7):1431–1439.

Barandas, M., Famiglini, L., Campagner, A., Folgado, D., Simão, R., Cabitza, F., and Gamboa, H. (2024). Evaluation of uncertainty quantification methods in multi-label classification: A case study with automatic diagnosis of electrocardiogram. *Information Fusion*, 101:101978.

Baumann, D., Kowalczyk, K., Rojas, C. R., Tiels, K., and Wachel, P. (2025). Safety and optimality in learning-based control at low computational cost. *IEEE Transactions on Automatic Control*, pages 1–13.

Baumann, D., Kowalczyk, K., Tiels, K., and Wachel, P. (2023). A computationally lightweight safe learn-

ing algorithm. In *IEEE Conference on Decision and Control*, pages 1022–1027.

Baumann, D. and Schön, T. B. (2024). Safe reinforcement learning in uncertain contexts. *IEEE Transactions on Robotics*, 40:1828–1841.

Brunke, L., Greeff, M., Hall, A. W., Yuan, Z., Zhou, S., Panerati, J., and Schoellig, A. P. (2022). Safe learning in robotics: From learning-based control to safe reinforcement learning. *Annual Review of Control, Robotics, and Autonomous Systems*, 5(1):411–444.

Buldygin, V. V. and Moskvichova, K. K. (2013). The sub-Gaussian norm of a binary random variable. *Theory of Probability and Mathematical Statistics*, 86:33–49.

Calliess, J.-P., Roberts, S. J., Rasmussen, C. E., and Maciejowski, J. (2020). Lazily adapted constant kinky inference for nonparametric regression and model-reference adaptive control. *Automatica*, 122:109216.

Deng, L. (2012). The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142.

Gal, Y. and Ghahramani, Z. (2016). Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *International Conference on Machine Learning*, pages 1050–1059.

Gao, J., Zhang, H., Lu, P., and Wang, Z. (2019). An effective LSTM recurrent network to detect arrhythmia on imbalanced ECG dataset. *Journal of Healthcare Engineering*, 2019:1–10.

Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. (2017). On Calibration of Modern Neural Networks. In *Proceedings of the 34th International Conference on Machine Learning*, pages 1321–1330. PMLR.

Hall, P. (1988). Theoretical comparison of bootstrap confidence intervals. *The Annals of Statistics*, 16(3):927–953.

Hedén, B., Ohlsson, M., Holst, H., Mjöman, M., Rittner, R., Pahlm, O., Peterson, C., and Edenbrandt, L. (1996). Detection of frequently overlooked electrocardiographic lead reversals using artificial neural networks. *The American Journal of Cardiology*, 78(5):600–604.

Huang, J. W., Roberts, S., and Calliess, J.-P. (2023). On the Sample Complexity of Lipschitz Constant Estimation. *Transactions on Machine Learning Research*.

Härdle, W. (2002). *Applied Nonparametric Regression*. Number 19 in Econometric Society Monographs. Cambridge Univ. Press, Cambridge, transferred to digital printing edition.

Jha, C. K. and Kolekar, M. H. (2020). Cardiac arrhythmia classification using tunable Q-wavelet transform based features and support vector machine classifier. *Biomedical Signal Processing and Control*, 59:101875.

Juditsky, A., Hjalmarsson, H., Benveniste, A., Delyon, B., Ljung, L., Sjöberg, J., and Zhang, Q. (1995). Nonlinear black-box models in system identification: mathematical foundations. *Automatica*, 31(12):1725–1750.

Kachuee, M., Fazeli, S., and Sarrafzadeh, M. (2018). ECG heartbeat classification: a deep transferable representation. In *IEEE International Conference on Healthcare Informatics*, pages 443–444.

Kowalczyk, K., Wachel, P., and Rojas, C. R. (2024). Kernel-Based Learning with Guarantees for Multi-agent Applications. In Franco, L., De Mulatier, C., Paszynski, M., Krzhizhanovskaya, V. V., Dongarra, J. J., and Sloot, P. M. A., editors, *Computational Science – ICCS 2024*, volume 14834, pages 479–487. Springer Nature Switzerland, Cham. Series Title: Lecture Notes in Computer Science.

Kumari, L. V. R. and Sai, Y. P. (2022). Classification of ECG beats using optimized decision tree and adaptive boosted optimized decision tree. *Signal, Image and Video Processing*, 16(3):695–703.

Ljung, L. (2006). Some aspects on nonlinear system identification. *IFAC Proceedings Volumes*, 39(1):110–121.

Longo, L., Brcic, M., Cabitza, F., Choi, J., Confalonieri, R., Ser, J. D., Guidotti, R., Hayashi, Y., Herrera, F., Holzinger, A., Jiang, R., Khosravi, H., Lecue, F., Malgieri, G., Páez, A., Samek, W., Schneider, J., Speith, T., and Stumpf, S. (2024). Explainable Artificial Intelligence (XAI) 2.0: A manifesto of open challenges and interdisciplinary research directions. *Information Fusion*, 106:102301.

Magureanu, S., Combes, R., and Proutiere, A. (2014). Lipschitz bandits: Regret lower bound and optimal algorithms. In *Conference on Learning Theory*, pages 975–999. PMLR.

Moody, G. and Mark, R. (2001). The impact of the MIT-BIH Arrhythmia Database. *IEEE Engineering in Medicine and Biology Magazine*, 20(3):45–50.

Muyskens, A., Priest, B., Goumiri, I., and Schneider, M. (2021). MuyGPs: Scalable Gaussian process hyperparameter estimation using local cross-validation. arXiv:2104.14581 [stat].

Mzyk, G. and Wachel, P. (2020). Wiener system identification by input injection method. *International Journal of Adaptive Control and Signal Processing*, 34(8):1105–1119.

Nadaraya, E. A. (1964). On estimating regression. *Theory of Probability and Its Applications*, 9(1):141–142.

Nadaraya, E. A. (1989). *Nonparametric Estimation of Probability Densities and Regression Curves*. Springer Netherlands, Dordrecht.

Nnyaba, U. V., Shemtaga, H. M., Collins, D. W., Muyskens, A. L., Priest, B. W., and Billor, N. (2024). Enhancing electrocardiography data classification confidence: A robust Gaussian process approach (MuyGPs). arXiv:2409.04642 [stat].

Novara, C., Fagiano, L., and Milanese, M. (2013). Direct feedback control design for nonlinear systems. *Automatica*, 49(4):849–860.

Powell, M. J. D. (1964). An efficient method for finding the minimum of a function of several variables without calculating derivatives. *The Computer Journal*, 7(2):155–162.

Prakasa Rao, B. L. S., editor (1983). *Nonparametric Functional Estimation*. Probability and Mathematical Statistics: A Series of Monographs and Textbooks. Academic Press.

Rao, N. and Protopopescu, V. (1996). On PAC learning of functions with smoothness properties using feedforward sigmoidal networks. *Proceedings of the IEEE*, 84(10):1562–1569.

Rasmussen, C. E. and Williams, C. K. I. (2008). *Gaussian Processes for Machine Learning*. Adaptive Computation and Machine Learning. MIT Press, Cambridge, Mass., 3. print edition.

Sattar, S., Mumtaz, R., Qadir, M., Mumtaz, S., Khan, M. A., De Waele, T., De Poorter, E., Moerman, I., and Shahid, A. (2024). Cardiac arrhythmia classification using advanced deep learning techniques on digitized ecg datasets. *Sensors*, 24(8).

Schuster, E. and Yakowitz, S. (1979). Contributions to the theory of nonparametric regression, with application to system identification. *The Annals of Statistics*, 7(1).

Schölkopf, B. and Smola, A. J. (2001). *Learning with kernels: Support vector machines, regularization, optimization, and beyond*. The MIT Press.

Sergeyev, Y. D. (1995). An information global optimization algorithm with local tuning. *SIAM Journal on Optimization*, 5(4):858–870.

Shahid, M. B., Robison, R., Shafer, T., Diloreto, V., Alexandrov, N. M., and Fleming, C. H. (2024). Uncertainty quantification using deep ensembles for decision making in cyber-physical systems. In *AIAA SCITECH Forum*, page 0108.

Shalev-Shwartz, S. and Ben-David, S. (2014). *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, Cambridge.

Strongin, R. G. (1973). On the convergence of an algorithm for finding a global extremum. *Engineering Cybernetics*, 11:549–555.

Tokmak, A., Krishnan, K. G., Schön, T. B., and Baumann, D. (2025). Safe exploration in reproducing kernel Hilbert spaces. In *International Conference on Artificial Intelligence and Statistics*.

Villacampa-Calvo, C., Zaldívar, B., Garrido-Merchán, E. C., and Hernández-Lobato, D. (2021). Multi-class Gaussian process classification with noisy inputs. *Journal of Machine Learning Research*, 22(36):1–52.

Watson, G. S. (1964). Smooth regression analysis. *Sankhyā: The Indian Journal of Statistics, Series A (1961-2002)*, 26(4):359–372.

Wood, G. R. and Zhang, B. P. (1996). Estimation of the Lipschitz constant of a function. *Journal of Global Optimization*, 8(1):91–103.

Zhang, W., Di, X., Wei, G., Geng, S., Fu, Z., and Hong, S. (2022). A deep Bayesian neural network for cardiac arrhythmia classification with rejection from ECG recordings. arXiv:2203.00512 [eess].

Zhou, F. and Fang, D. (2024). Multimodal ECG heartbeat classification method based on a convolutional neural network embedded with FCA. *Scientific Reports*, 14(1):8804.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes]
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes]
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes]

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Yes]
   (b) Complete proofs of all theoretical results. [Yes]
   (c) Clear explanations of any assumptions. [Yes]

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes]
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes]
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes]
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Not Applicable]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [Yes]
   (b) The license information of the assets, if applicable. [Not Applicable]
   (c) New assets either in the supplemental material or as a URL, if applicable. [Yes]
   (d) Information about consent from data providers/curators. [Not Applicable]
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. [Not Applicable]
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

Computationally Lightweight Classifiers with Frequentist Bounds on Predictions: Appendix

Contents

A BOUNDS ON THE BIAS **14**

A.1 Proof of Lemma 1 14

A.2 Proof of Lemma 2 15

A.3 Extension to positive definite kernels with infinite support 15

B BOUNDS ON THE SAMPLING ERROR **16**

B.1 Concentration inequality for self-normalized sums with sub-Gaussian noise 16

B.2 Proof of Lemma 4 16

C LIPSCHITZ-CONTINUOUS SYNTHETIC DATASET GENERATION **18**

D ESTIMATING A LIPSCHITZ CONSTANT FROM DATA **18**

E SUPPLEMENTARY RESULTS **19**

E.1 Additional figures for dyadic classifier predictions 19

E.2 Additional results from the ECG dataset 19

E.3 Results on the MNIST dataset 20

E.4 Results on a simplified GTSRB dataset 20

E.5 Comparison with other baselines 20

F ABLATIONS **21**

A BOUNDS ON THE BIAS

In this section, we restate and prove Lemmas 1, 2, and 3 from Section 3.2.

A.1 Proof of Lemma 1

Lemma 1. *Under Assumptions 1 and 4, we have, for all $n \geq 0$ and $y \in \mathcal{Y}$,*

$$|p_c(y) - \bar{p}_c(y)| \leq L\lambda, \tag{8}$$

where L is the known Lipschitz constant from (2) and λ is the user-defined kernel bandwidth from (6).

Proof. For notational convenience, we denote the kernel weight of the estimator at each iteration, $K_\lambda(y, y_i)/\kappa_n(y)$, from (5) by θ_i. Notably, these weights sum to 1. The virtual estimate $\bar{p}_c(y)$ in the bias term can be represented as the weighted sum of the true probability at each observation:

$$|p_c(y) - \bar{p}_c(y)| = \left| p_c(y) - \sum_{t=1}^{n} \theta_i p_c(y_i) \right|. \tag{16}$$

Furthermore, since the weights sum to 1, the term $p_c(y)$ can be introduced into a summation term without altering its value:

$$p_c(y) = p_c(y) \sum_{i=1}^{n} \theta_i = \sum_{i=1}^{n} \theta_i p_c(y). \tag{17}$$

Equation (17) can be used to rewrite and simplify (16) as

$$\begin{aligned} |p_c(y) - \bar{p}_c(y)| &= \left| \sum_{i=1}^{n} \theta_i p_c(y) - \sum_{i=1}^{n} \theta_i p_c(y_i) \right| \\ &= \left| \sum_{i=1}^{n} \theta_i (p_c(y) - p_c(y_i)) \right|. \end{aligned}$$

By the Cauchy-Schwarz inequality,

$$\left| \sum_{i=1}^{n} \theta_i (p_c(y) - p_c(y_i)) \right| \leq \sum_{i=1}^{n} \theta_i \left| p_c(y) - p_c(y_i) \right|. \tag{18}$$

However, under Assumption 1, the term $|p_c(y) - p_c(y_i)|$ is bounded by a known, finite Lipschitz constant L as

$$|p_c(y) - p_c(y_i)| \leq L\|y - y_i\|. \tag{19}$$

Furthermore, due to Assumption 4, if $K(y, y_i) \geq 0$, then $\theta_i \geq 0$, and $\|y - y_i\| \leq \lambda$. Therefore, by combining (18) and (19), we obtain

$$\begin{aligned} |p_c(y) - \bar{p}_c(y)| &\leq \sum_{i=1}^{n} \theta_i \left| p_c(y) - p_c(y_i) \right| \\ &\leq \sum_{i=1}^{n} \theta_i L\|y - y_i\| \leq L\lambda. \end{aligned} \tag{20}$$

□

A.2 Proof of Lemma 2

Lemma 2. *Under Assumptions 2 and 4, we have for almost every $y \in \mathcal{Y}$*

$$|p_c(y) - \bar{p}_c(y)| \leq \frac{\lambda}{\gamma}, \tag{9}$$

where λ is the user-defined kernel bandwidth parameter as depicted in (6) and γ, in accordance with Assumption 2, is the known margin of the distribution $\mathcal{D}$.

Proof. Due to Assumption 2, $p_c(y)$ is equivalent to a function $f : \mathcal{Y} \to [0, 1]$ with Lipschitz constant $1/\gamma$ for almost all $y \in \mathcal{Y}$. This would imply that $\mathbb{1}_{c_i} = f(y_i)$ with probability 1. For example, if we define a set $\mathcal{R}$

$$\mathcal{R} \coloneqq \{y \in \mathcal{Y} : p_c(y) = 0\},$$

we can choose f as

$$f(y) = \min\left\{1, \frac{1}{\gamma} \inf_{y' \text{ in } \mathcal{R}} \|y - y'\|\right\}.$$

From here, the proof proceeds identically as that of Lemma 1 from Appendix Section A.1. The terms $p_c(y)$ from (19) could be replaced with $f(y)$, following which the result from (20) would remain valid for almost all $y \in \mathcal{Y}$. □

A.3 Extension to positive definite kernels with infinite support

Assumption 4 limits kernel choice to those with support only in $[-1, 1]$. While this can easily be accomplished for any kernel by truncating its outputs to zero outside the interval $[-1, 1]$, this aspect of Assumption 4 can be relaxed by assuming bounded inputs. In this case, we can extend our results to positive definite kernels with infinite support. Doing so would conservatively expand Lemmas 1 and 2 to accommodate positive definite kernels with infinite support.

Lemma 3. *Under Assumptions 1, 2, 4 with the change that $K(v) \geq 0$ for all $\|v\| > 1$, and 5 we have for almost every $y \in \mathcal{Y}$*

$$|p_c(y) - \bar{p}_c(y)| \leq \beta\lambda^* + \beta\Phi\varepsilon_t, \tag{11}$$

where

$$\varepsilon_t \coloneqq \sum_{i \in \mathcal{I}_{\text{far}}} \frac{K_\lambda(y, y_i)}{\kappa_n(y)} \|y - y_i\|$$

is a term that represents the weighted sum corresponding to the samples in the tail of the kernel's span (indices $\mathcal{I}_{\text{far}}$, where $\|y - y_i\| > \lambda^*$), Φ is the input space diameter from Assumption 5, and $\beta = L$ or $1/\gamma$ depending on whether we assume an overlapping (see Assumption 1) or a separable (see Assumption 2) distribution on $\mathcal{D}$.

Proof. We introduce a cut-off radius r^* such that and use the Cauchy-Schwarz inequality (similar to (20)) to split $|p_c(y) - \bar{p}_c(y)|$ into two terms, with indices in $\mathcal{I}_{\text{near}}$ for instances where $\|y - y_i\| \leq \lambda^*$, and $\mathcal{I}_{\text{far}}$ corresponding to indices where $\|y - y_i\| > \lambda^*$, where $r^* = \lambda^*/\lambda$. We can write that split as

$$|p_c(y) - \bar{p}_c(y)| \leq \beta \sum_{i \in \mathcal{I}_{\text{near}}} \theta_i \|y - y_i\| + \beta \sum_{i \in \mathcal{I}_{\text{far}}} \theta_i \|y - y_i\|. \tag{21}$$

Since the kernel weights sum to 1, samples in the near set $\mathcal{I}_{\text{near}}$ are bounded by λ^* as

$$\beta \sum_{i \in \mathcal{I}_{\text{near}}} \theta_i \|y - y_i\| \leq \beta\lambda^* \sum_{i \in \mathcal{I}_{\text{near}}} \theta_i \leq \beta\lambda^*. \tag{22}$$

For samples in the $\mathcal{I}_{\text{far}}$ set, the distances and weights can be large, but they are capped by Assumption 5:

$$\sum_{i \in \mathcal{I}_{\text{far}}} \theta_i \|y - y_i\| \leq \Phi \sum_{i \in \mathcal{I}_{\text{far}}} \theta_i. \tag{23}$$

We set $\varepsilon_t \coloneqq \sum_{i \in \mathcal{I}_{\text{far}}} \theta_i$, to obtain the bounds

$$|p_c(y) - \bar{p}_c(y)| \leq \beta\lambda^* + \beta\Phi\varepsilon_t, \tag{24}$$

where $\beta = L$ or $1/\gamma$. □

The choice of kernel heavily influences the practical utility of the bounds from (24). For a standard RBF kernel, setting $r^* \coloneqq 3$ captures 99.7% of the kernel's mass. Yet, in large datasets, the number of points in the $\mathcal{I}_{\text{far}}$ set vastly outnumber those in the $\mathcal{I}_{\text{near}}$ set. This would cause the tail term $\beta\Phi\varepsilon_t$ to blow up, resulting in the bounds being unreasonably conservative.

B BOUNDS ON THE SAMPLING ERROR

In this section, we restate and prove Lemma 4 from Section 3.2.2.

B.1 Concentration inequality for self-normalized sums with sub-Gaussian noise

Here, we restate a result from Baumann et al. (2023, 2025) which we use to prove Lemma 4. This lemma introduces a concentration inequality for self-normalized sums with sub-Gaussian noise (Abbasi-Yadkori et al., 2011, Thm 3).

Lemma 5 (Baumann et al. (2023, 2025))**.** *Let $\{v_t : t \in \mathbb{N}\}$ be a bounded stochastic process and $\{\omega_t : t \in \mathbb{N}\}$ be an i.i.d. sub-Gaussian stochastic process, i.e., there exists a $\sigma > 0$ such that, for any $\rho \in \mathbb{R}$, and any $t \in \mathbb{N}$,*

$$\mathbb{E}[\exp(\rho\omega_t)] \leq \exp\left(\frac{\rho^2\sigma^2}{2}\right).$$

Further, let $S_n \coloneqq \sum_{t=1}^{n} v_t\omega_t$ and $V_n \coloneqq \sum_{t=1}^{n} v_t^2$. Then for any $n \in \mathbb{N}$ and $0 < \delta < 1$, with probability at least $1 - \delta$,

$$|S_n| \leq \sqrt{2\sigma^2 \log(\delta^{-1}\sqrt{1 + V_n})(1 + V_n)}. \tag{25}$$

Correspondingly, we derive bounds on the sampling error. In this section, we utilize Lemma 5 to prove Lemma 4.

B.2 Proof of Lemma 4

Lemma 4. *Under Assumptions 3 and 4, we have, for all $n \geq 0$, with probability at least $1 - \delta$,*

$$|\bar{p}_c(y) - \hat{p}_c(y)| \leq 2\sigma \frac{\alpha_n(y, \delta)}{\kappa_n(y)}, \tag{12}$$

where

$$\alpha_n(y, n) = \begin{cases} \sqrt{\kappa_n(y) \log(\delta^{-1}\sqrt{1 + \kappa_n(y)})}, \\ \qquad\qquad \text{if } \kappa_n(y) > 1, \\ \sqrt{\log\left(\sqrt{2}/\delta\right)}, \\ \qquad\qquad \text{if } 0 < \kappa_n(y) \leq 1. \end{cases} \tag{13}$$

Proof. Using the same notation for normalized weights (θ_i) as Appendix A.1, the difference between the virtual estimate $\bar{p}_c(y)$ and the actual estimate $\hat{p}_c(y)$ can be expressed as

$$|\bar{p}_c(y) - \hat{p}_c(y)| = \left| \sum_{i=1}^{n} \theta_i (p_c(y) - \mathbb{1}_{c_i}(c)) \right|. \tag{26}$$

The first term $\sum_{i=1}^{n} \theta_i p_c(y)$ is the virtual estimate the classifier in (5) would produce if it had true probability labels rather than discrete labels. Here, for one-hot vectorized class label representations $\mathbb{1}_{c_i} := \mathbb{1}\{c_i = c\}$, $\mathbb{1}_{c_i}(c)$ represents the indicator function value at class index c. For convenience in notations, we denote the error between true and discrete labels $p_c(y) - \mathbb{1}_{c_i}(c)$ as ε_i.

Since c is a Bernoulli random variable with success probability q_c, the random variable $q_c - c$ is σ-sub-Gaussian with $\sigma \leq 1/4$ (Buldygin and Moskvichova, 2013, Thm 2.1 and Lemma 2.1). Expanding this scalar result into the $\mathbb{R}^d$ problem setting, the context label c is derived from the indicator function $\mathbb{1}_{c_i}(c)$ and the success probabilities $p_c(y)$. Therefore, the term ε_i can be conceived of as an i.i.d. noise term that is σ-sub-Gaussian with $\sigma \leq 1/4$. With this observation, we can frame θ_i as a stochastic process since its values are always lesser than 1, and ε_i can be conceptualized as an i.i.d. σ-sub-Gaussian stochastic process.

Consequently, the term $\sum_{i=1}^{n} \theta_i \varepsilon_i$ can be likened to S_n from Lemma 5 and $\sum_{i=1}^{n} \theta_i^2$ to V_n.

Expanding the θ_i notation out into its original form, we see that the term

$$\left| \sum_{i=1}^{n} \theta_i \varepsilon_i \right| = \frac{1}{\kappa_n(y)} \left| \sum_{i=1}^{n} K_\lambda^2(y, y_i) \varepsilon_i \right|$$

is upper bounded with probability at least $1 - \delta$ by

$$\frac{\sigma}{\kappa_n(y)} \sqrt{2 \log \left(\frac{1}{\delta} \sqrt{1 + \sum_{i=1}^{n} K_\lambda^2(y, y_i)} \right) \left(1 + \sum_{i=1}^{n} K_\lambda^2(y, y_i) \right)}. \tag{27}$$

Furthermore, since $K_\lambda(y, y_i) \leq 1$ (see Assumption 4), it follows that $K_\lambda^2(y, y_i) \leq K_\lambda(y, y_i)$. This allows us to upper bound the summation term as

$$\sum_{i=1}^{n} K_\lambda^2(y, y_i) \leq \sum_{i=1}^{n} K_\lambda(y, y_i).$$

The term on the right is, by definition, $\kappa_n(y)$. Substituting this into the bound, we obtain

$$\frac{1}{\kappa_n(y)} \left| \sum_{i=1}^{n} K_\lambda^2(y, y_i) \varepsilon_i \right| \leq \sigma \sqrt{2 \log(\delta^{-1} \sqrt{1 + \kappa_n(y)}} \frac{\sqrt{1 + \kappa_n(y)}}{\kappa_n(y)}.$$

Additionally, we can observe that if $\kappa_n(y) > 1$, then

$$\frac{\sqrt{1 + \kappa_n(y)}}{\kappa_n(y)} < \frac{\sqrt{2\kappa_n(y)}}{\kappa_n(y)} = \frac{\sqrt{2}}{\sqrt{\kappa_n(y)}}.$$

Therefore, with probability at least $1 - \delta$, for $\kappa_n(y) > 1$,

$$\frac{1}{\kappa_n(y)} \left| \sum_{i=1}^{n} K_\lambda^2(y, y_i) \varepsilon_i \right| \leq \frac{2\sigma}{\kappa_n(y)} \sqrt{\kappa_n(y) \log(\delta^{-1} \sqrt{1 + \kappa_n(y)})}, \tag{28}$$

and for $0 < \kappa_n(y) \leq 1$,

$$\frac{1}{\kappa_n(y)} \left| \sum_{i=1}^{n} K_\lambda^2(y, y_i) \varepsilon_i \right| \leq \sigma \sqrt{2 \log \left(\delta^{-1} \sqrt{1 + \kappa_n(y)} \right)} \frac{\sqrt{1 + \kappa_n(y)}}{\kappa_n(y)}. \tag{29}$$

We can see here that if $\kappa_n(y) \leq 1$, then the term $\sqrt{1+\kappa_n(y)} \leq \sqrt{2}$. By taking the worst-case bounds, we can thus simplify the term in (29) to

$$\sigma\sqrt{2\log\left(\delta^{-1}\sqrt{1+\kappa_n(y)}\right)}\frac{\sqrt{1+\kappa_n(y)}}{\kappa_n(y)} \leq \frac{2\sigma}{\kappa_n(y)}\sqrt{\log\left(\frac{\sqrt{2}}{\delta}\right)}. \tag{30}$$

By collecting the terms from (28) and (30), we arrive at

$$|\bar{p}_c(y) - \hat{p}_c(y)| \leq \begin{cases} \frac{2\sigma}{\kappa_n(y)}\sqrt{\kappa_n(y)\log(\delta^{-1}\sqrt{1+\kappa_n(y)})} & \text{if } \kappa_n(y) > 1, \\ \frac{2\sigma}{\kappa_n(y)}\sqrt{\log\left(\frac{\sqrt{2}}{\delta}\right)} & \text{if } 0 < \kappa_n(y) \leq 1. \end{cases} \tag{31}$$

□

C LIPSCHITZ-CONTINUOUS SYNTHETIC DATASET GENERATION

In this section, we provide details on how we generated synthetic data (see Section 4) to match Assumption 1. We define an underlying probability function $p_c(y, L)$ that is a function of a known Lipschitz constant L. Instead of generating hard-labelled measurements, we sample data points according to this function. This is easy to achieve in a case of binary classification, where the relationship between probabilities is exact. That is, for two classes, we have a single separating hyperplane. The probability for a sample y can be modelled using the logistic (sigmoid) function, which takes the signed distance to the hyperplane as input. The probability p_c for class $c = 1$ would be

$$p_c(y) = \frac{1}{1+\exp(-k(wy+b))}, \tag{32}$$

where w is the normal vector to the hyperplane with $\|w\| = 1$ and k is a scaling factor that controls the steepness of the probability function. The gradient of this function,

$$\nabla p_1(y) = k \cdot p_1(y) \cdot (1 - p_1(y)) \cdot w, \tag{33}$$

is maximized at $p_1(y) = 0.5$, which leads to a direct relationship that $L = k/4$.

D ESTIMATING A LIPSCHITZ CONSTANT FROM DATA

The Lipschitz constant L in Assumption 1 and the equivalent margin in Assumption 2 are parameters that convey the underlying assumptions about the smoothness of the data. It influences the strength of the classifier's bounds. However, since the underlying probability function $p_c(y)$ cannot be measured, we must estimate this constant from data. Correspondingly, it is also important to first determine if the dataset falls into the category of overlapping or separable distributions.

To approximate the Lipschitz constant from data, the maximum pairwise distance forms a basis for an upper bound. Here, we assume that for two samples y and y' with matching labels that are the closest to each other in the entire dataset, the true probability $p_c(y)$ and $p_c(y')$ do not differ by more than a threshold probability P_t for any $c \in \mathcal{C}$. Then, we can estimate the Lipschitz constant from (2) as

$$L \geq \frac{P_t}{\sup\{\|y - y'\|\}}. \tag{34}$$

Lastly, to determine the nature of the data distribution, we selected 1000 instances from the MNIST and MIT-BIH dataset with random stratified sampling. We computed the pairwise distances from each sample to all other samples in this set. We observed that the maximum pairwise distance within a class was comparable to the maximum pairwise distance observed globally. For both datasets, this implies that their distributions are overlapping, rather than being separated by a margin. Additionally, in Figure 10, we show $K(v)$ for all pairs of a random selection of 100 samples from the MNIST dataset for its optimized bandwidth $\lambda = 7.5$.

In Remark 1 (Section 2), we introduce various approaches in the literature that have been used to estimate Lipschitz constants from measurements. Most methods are heuristic, including the one we use based on Strongin (1973). Estimating an upper bound on the smoothness of a function with guarantees is intractable without invoking further regularity assumptions. This is done so in the form of assuming bounded higher-order derivatives (Huang et al., 2023), or by invoking sampling assumptions about the measurements Tokmak et al. (2025). The former is best suited for approaches where it is more reliable to know a limit on the second-order derivative: for example, a motor is bounded by its maximum torque. On the other hand, the latter approach is better in cases where we can independently sample from a family of functions in a probability space: for example, an experimentally determined noise oracle. In our case, $p_c(y)$ is entirely unknown; we neither have knowledge of a bounded higher-order derivative nor a reliable class of functions from a probability space to sample from. Therefore, in this paper, we approximate the Lipschitz constant from available data.

E SUPPLEMENTARY RESULTS

In this section, we report further details from the experiments conducted in Section 4 and also provide further experimental results.

E.1 Additional figures for dyadic classifier predictions

In the *dyadic* approach, we construct a hash table that maps the grid index to the aggregated class count vector. Figure 4 shows two synthetic datasets corresponding to Assumptions 1 and 2, and their respective hash tables marked with majority predictions.

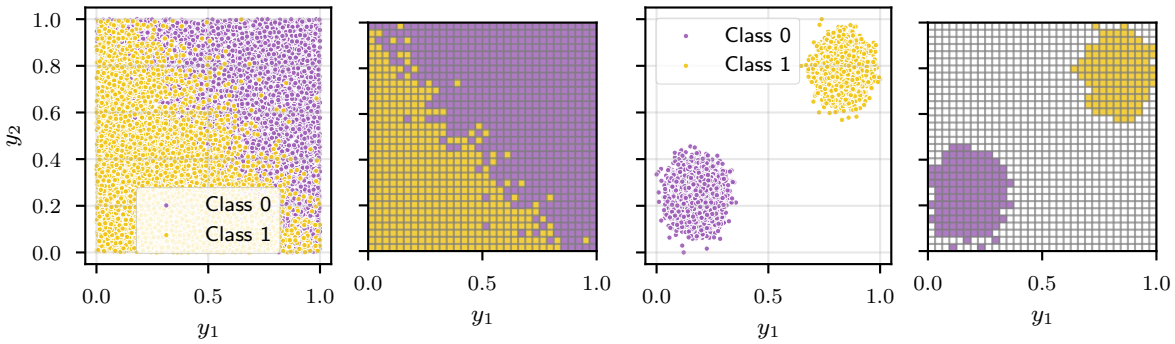


Figure 4: Two synthetic datasets and their corresponding dyadic cell prediction grids. *The figures on the left correspond to the Lipschitz-continuous overlapping dataset; the figures on the right correspond to the dataset separated by a margin.*

E.2 Additional results from the ECG dataset

In this section, we show additional results from the experiments conducted in Section 4 on the ECG dataset. Figure 5 shows randomly selected ECG waveforms from the testing set and their associated class prediction probabilities. In Figure 6 we show the alignment between prediction confidence ($\max \hat{p}_c$) and accuracy. This can be used to compute the ECE (Guo et al., 2017), which in our case is 7.5%.

Next, following the conventions of Nnyaba et al. (2024), we define a Type I error as the instance where the classifier's prediction, i.e., $\max \hat{p}_c$, is incorrect, and a Type II error as an instance where the intervals are so wide that $\hat{p}_c$ could be less than 0.5. In Figure 7 we show the Type I and Type II error counts obtained from the *regular* classifier's predictions on the entire testing set. Here, we also see that the errors are overrepresented in the minority classes. If we count Type II errors as incorrect predictions, the accuracy of our classifier is reduced to 84%.

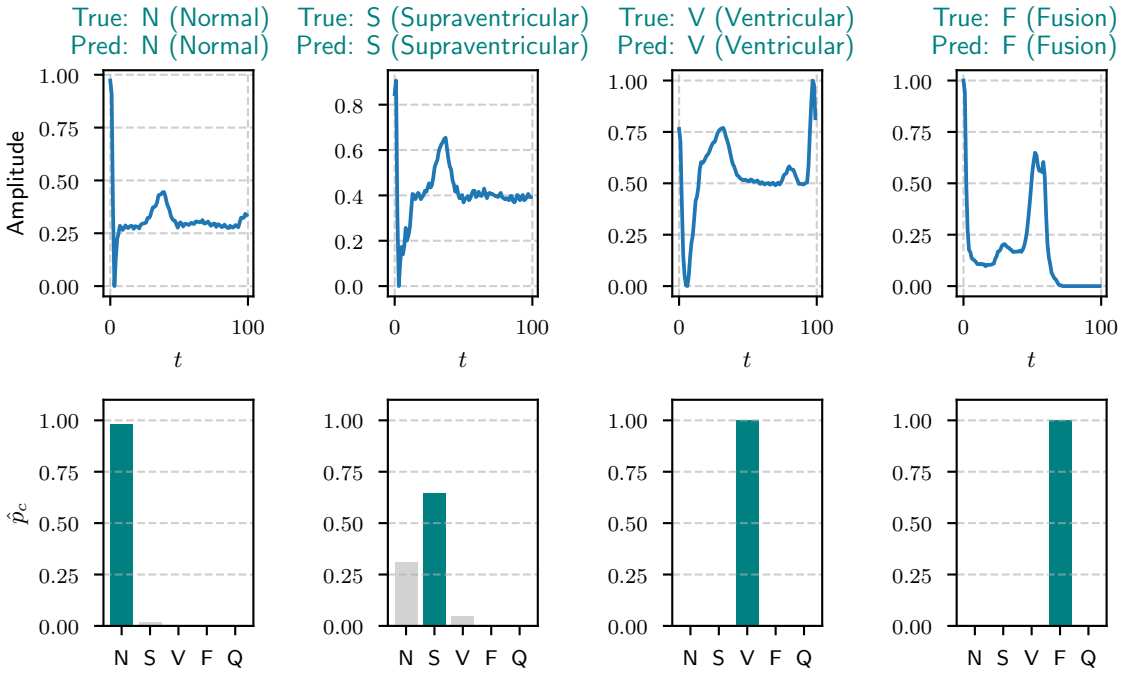


Figure 5: Illustrative ECG waveforms (amplitude normalized from mV, t denotes time step) and associated class probabilities for each class from the MIT-BIH database; class Q (unclassifiable) has been excluded.

E.3 Results on the MNIST dataset

In addition to synthetic and ECG datasets, we implemented the proposed classifier on the MNIST handwritten digits dataset (Deng, 2012). The dataset contains 28×28 pixel black-and-white images of handwritten digits commonly used for training various image processing classifiers. The dataset contains 60 000 training samples split equally among classes and 10 000 training samples. Figure 8 shows predictions, accuracy trends, and bound trends for the *regular* classifier trained on the MNIST dataset with $L = 0.03$ and $\lambda = 7.5$. From the figure, we observe that the classifier demonstrates an accuracy of >92% on the dataset with strong bounds with just 10 000 samples.

E.4 Results on a simplified GTSRB dataset

The improved computational efficiency of the Nadaraya-Watson estimator is of significant advantage when applied to higher-dimensional systems. While we overcome the computational bottleneck associated with GP-like methods, we are still limited by how expressive distances can be in higher dimensional spaces. These limits are best explored and tested in the context of high-dimensional RGB images. We evaluate our *regular* and *localized* classifier on a subset of the German Traffic Sign Recognition Benchmark (GTSRB) dataset. We simplify to a binary classification problem including only the 'stop' sign and 'end of speed limit' sign. Figure 9 shows these results with $L = 0.05$, $\lambda = 7.5$, and $k = 20$ for the *localized* classifier.

E.5 Comparison with other baselines

Tasked with ECG heartbeat classification, various classical and deep learning methods have achieved highly accurate results on the MIT-BIH database. Our contribution lies not only in achieving high accuracy scores but also in providing actionable uncertainty quantification in relation to the classifier's predictions. These theoretical guarantees are often entirely absent in such methods. Nevertheless, in this section, we summarize and review the accuracy obtained by such methods (those cited in Section 5) to contextualize our work's performance with respect to existing methods. For a more extensive comparison, we refer the reader to Table 1 and Table 2 from Sattar et al. (2024).

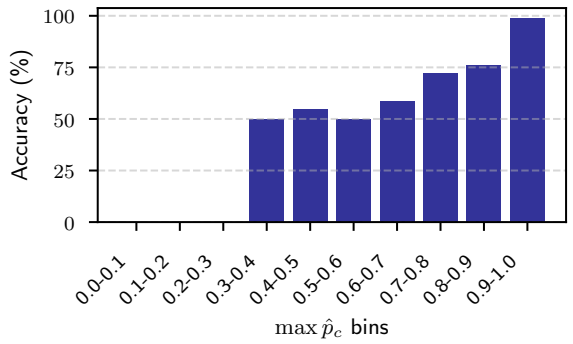


Figure 6: Accuracy of the *regular* NWC grouped by $\max \hat{p}_c$ bins. *The figure shows how the classifier's prediction confidence correlate with its accuracy.*

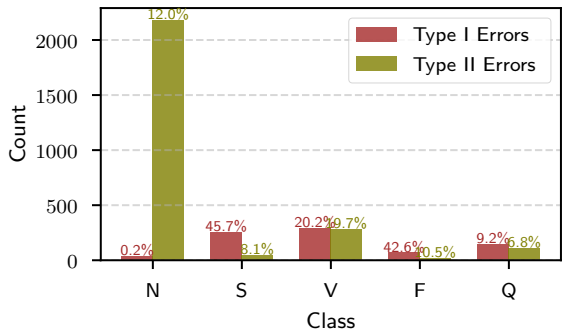


Figure 7: Type I and Type II error counts for each class for the *regular* classifier's predictions over the entire testing set. *The inset text also shows the percentage of the dataset these counts represent.*

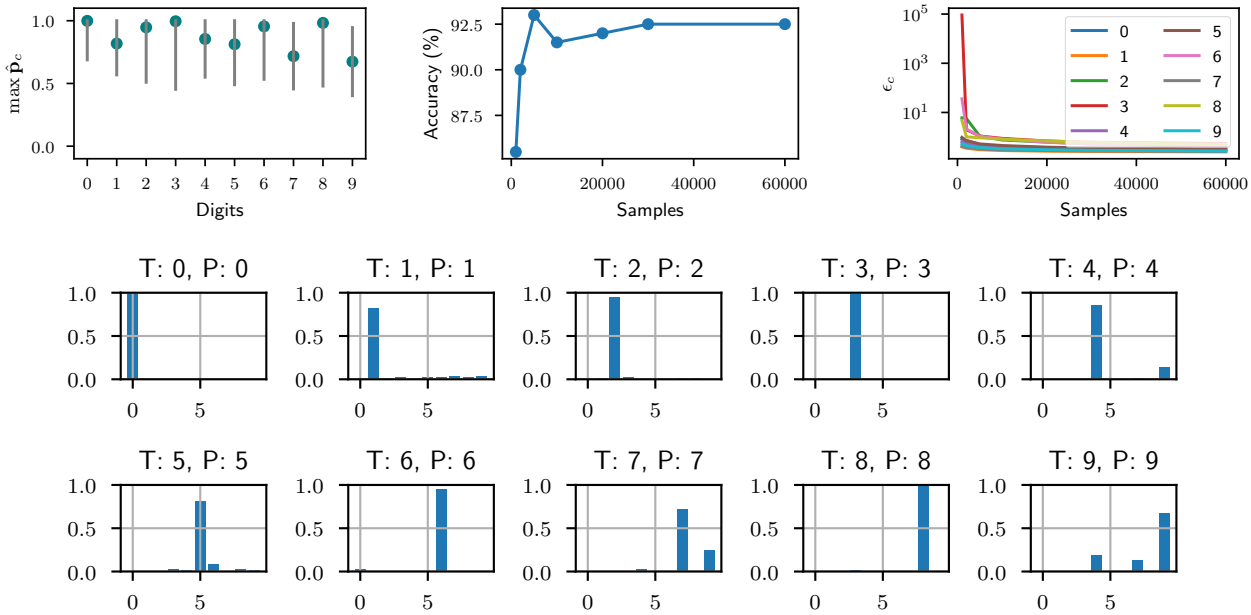


Figure 8: Results of the regular classifier on the MNIST dataset. *On the top row, we see predictions (left) and tightness of the bounds for 10 randomly selected samples of each digit from the dataset. Their corresponding all-class prediction probabilities are shown in the figure below, where true and predicted labels are marked with 'T' and 'P' respectively. Next on the top row, we show the accuracy (middle) and strength of the bounds (right) of the classifier with varying number of samples n.*

F ABLATIONS

In this section, we perform ablation studies on kernel choice and bandwidth parameter λ. As the Nadaraya-Watson estimator is a non-parametric kernel regressor, we study the effect different kernel functions have on the nature of the estimate. Table 4 lists the different kernel functions we evaluate our classifier with. Correspondingly, for each kernel, we optimize to find the bandwidth λ that maximizes the classifier's accuracy and bounds.

To this end, we define a weighted objective function that tries to balance accuracy A and average uncertainty

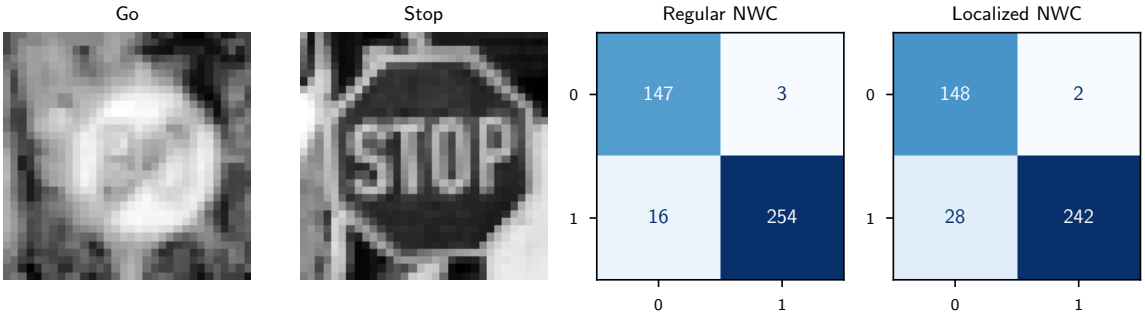


Figure 9: Results from the reduced GTSRB dataset. *On the left are the two classes considered in our study. On the right are confusion matrices from the two proposed variants of our classifier. In the confusion matrices, the label 0 corresponds to the 'go' sign and 1 corresponds to the 'stop' sign.*

Table 3: Summary of reported ECG classification results in the literature.

Literature	**Remark**	**Accuracy**	**Bounds**
Kumari and Sai (2022)	Optimized decision tree	98.77%	✗
Zhou and Fang (2024)	CNN with frequency channel attention	99.6%	✗
Abdalla et al. (2020)	CNN with 11 layers	99.84%	✗
Gao et al. (2019)	Long short-term memory model with focal loss	99.26%	✗
Jha and Kolekar (2020)	SVM with tunable Q-wavelet transform	99.27%	✗
Nnyaba et al. (2024)	GP classification with bounds	$\approx$ 98%	✓

interval B as a function of a user-defined weight r:

$$J(A, B) = rA - (1 - r)B. \tag{35}$$

Here, we utilize a simple weighted-sum that awards a higher score if A is maximized and B is minimized. The weight r directly quantifies the trade-off between the two potentially competing measures.

The optimization experiments were conducted on the MNIST dataset. We trained the *regular* NWC on 30 000 samples of the MNIST dataset with all the kernels from Table 4. We set the weight r in (35) as 0.95. Since A and B are experimentally computed, there is no direct way to find derivatives of J with respect to λ; thus, we used Powell's conjugate direction method to optimize for the score J (Powell, 1964). The results from Figure 10 show that the Epanechnikov kernel offered the best performance. Yet, as we see from the y-axis of the figure, the improvement difference was marginal. This aligns with previous research that have shown the limited effect of kernel choice on the Nadaraya-Watson estimator (Härdle, 2002).

Table 4: Different kernel functions and their definitions.

Kernel	**Definition**
Boxcar kernel	$K(v) = \begin{cases} 1 & \text{if } \lVert v \rVert \leq 1 \\ 0 & \text{otherwise} \end{cases}$
Gaussian kernel (without truncation)	$K(v) = \exp\left(-\frac{\lVert v \rVert^2}{2}\right)$
Epanechnikov kernel	$K(v) = \begin{cases} 1 - v^2 & \text{if } \lVert v \rVert \leq 1 \\ 0 & \text{otherwise} \end{cases}$
Quartic kernel	$K(v) = \begin{cases} (1 - v^2)^2 & \text{if } \lVert v \rVert \leq 1 \\ 0 & \text{otherwise} \end{cases}$
Triweight kernel	$K(v) = \begin{cases} (1 - v^2)^3 & \text{if } \lVert v \rVert \leq 1 \\ 0 & \text{otherwise} \end{cases}$
Tricube kernel	$K(v) = \begin{cases} (1 - \lvert v \rvert^3)^3 & \text{if } \lVert v \rVert < 1 \\ 0 & \text{otherwise} \end{cases}$
Cosine kernel	$K(v) = \begin{cases} \frac{\pi}{4} \cos\left(\frac{\pi}{2} v\right) & \text{if } \lVert v \rVert \leq 1 \\ 0 & \text{otherwise} \end{cases}$

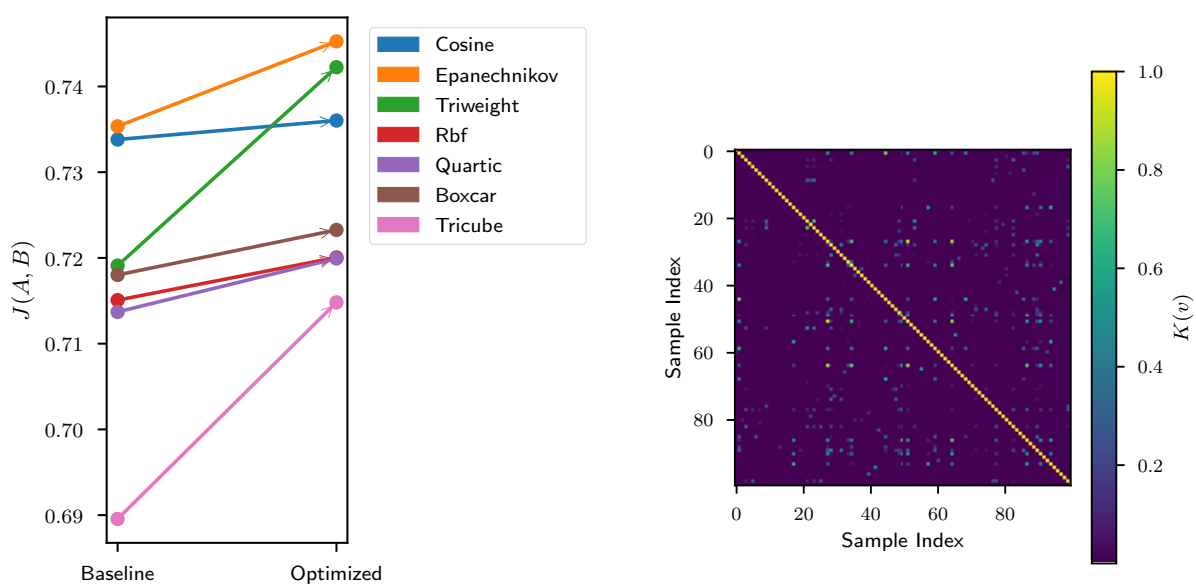


Figure 10: Hyperparameter optimization on the kernels. *On the left, we show the baseline and optimized score $J(A, B)$ on all kernels from Table 4. Here, the results show that the effect of hyperparameter optimization to find the optimal λ on each kernel is, at best, only moderately significant. On the right, we show the Epanechnikov kernel evaluated at all pairs of 100 randomly sampled MNIST images with optimized $\lambda = 7.5$.*