The Robustness of SVM Kernels to Noise:

A Comparative Analysis on MNIST

Alexander Eriksson Byström Jack Andersson Stridh

Abstract

This paper compares the robustness of linear, polynomial, and radial basis function (RBF) kernels to additive Gaussian noise on the MNIST dataset. We train SVMs with each kernel on clean data, then evaluate accuracy as noise intensity increases from 0% to 100% of the normalized pixel range. While the RBF kernel achieves the highest accuracy on clean images (96.50%), its performance degrades most rapidly under noise, falling to 19.07% at maximum intensity. The polynomial kernel demonstrates the strongest robustness, maintaining 39.95% accuracy at 100% noise compared to 27.35% for linear and 19.07% for RBF. We attribute this pattern to the bias-variance tradeoff: the RBF kernel's localized decision boundaries become unstable when noise disrupts fine-grained distance information, while the polynomial kernel's global structure allows perturbations to partially cancel across dimensions.

1 Introduction

Handwritten digit recognition has long served as a benchmark problem in machine learning, with the MNIST dataset becoming the standard testbed for evaluating classification algorithms. Modern deep learning methods now achieve near-perfect accuracy on clean MNIST data, but classical methods such as SVMs remain relevant in settings where computational resources are constrained or where interpretability is desired. And since real-world images inevitably contain some degree of noise from low-quality cameras, compression artifacts or poor lighting, robustness in the face of such perturbations is a practical concern.

Support Vector Machines provide a useful framework for studying noise robustness because of their geometric transparency. The maximum margin principle offers an intuitive basis for thinking about stability: observations far from the decision boundary should tolerate small perturbations better than those near it. The kernel function shapes this geometry, governing both the complexity of the decision surface and the dimensionality of the implicit feature space.

Yet the empirical literature on kernel robustness is surprisingly thin and somewhat contradictory. Some work suggests that simpler linear kernels generalize better as noise increases, which fits standard bias-variance intuitions. Other studies have found RBF kernels more robust on high-dimensional image data, though these results largely concern adversarial perturbations rather than random noise. It remains unclear whether the conclusions carry over.

This paper compares the robustness of linear, polynomial, and RBF kernels to Gaussian noise on MNIST. We progressively increase noise intensity and track how accuracy degrades for each kernel, aiming to identify which offers the best tradeoff between clean-data performance and noise tolerance. By focusing on stochastic rather than adversarial perturbations, we address a gap in the literature and provide evidence on whether kernel complexity helps or hurts under more realistic conditions.

2 Related Work

2.1 The SVM Framework

The theoretical foundations of Support Vector Machines emerged through a series of developments spanning three decades. Vapnik and Chervonenkis (1964) introduced the algorithm for finding maximum margin linear classifiers, establishing the core principle of maximizing the geometric margin between classes. However, this original formulation was limited to linearly separable data, restricting its practical applicability.

A crucial theoretical breakthrough came the following year when Aizerman, Braverman, and Rozonoer (1964) introduced what would later become known as the ”kernel trick.” They demonstrated that so-called ”potential functions”, now called kernels, could implicitly map data into higher-dimensional spaces, enabling non-linear decision boundaries without explicitly computing the transformed feature vectors. This insight would prove essential for the later development of modern SVMs. Complementing this work, Cover (1965) provided the theoretical justification by proving that non-linearly

separable data can always be correctly classified using separating hyperplanes when projected into a sufficiently high-dimensional space.

The modern SVM emerged through two final contributions. Boser, Guyon, and Vapnik (1992) synthesized these earlier ideas by applying the kernel trick to maximum margin classifiers, creating a practical algorithm for non-linear classification. The remaining limitation was sensitivity to noise: a single outlier could dramatically shift the decision boundary. Cortes and Vapnik (1995) addressed this by introducing soft-margin classification, which permits some misclassifications through slack variables while penalizing them in the objective function. The observations that lie on or within the margin boundaries, termed "support vectors," fully determine the classifier, making the method both computationally efficient and relatively robust to outliers and instances of noise contained in the interior of each class.

2.2 Empirical Findings

The empirical evidence on the robustness of various SVM kernels to noise is mixed and surprisingly sparse. Ljunggren and Ishii (2021) find that an SVM with a linear kernel is more robust to intentional data corruption than a radial or polynomial kernel, with the linear model actually outperforming the other two when noise levels are very high. This aligns clearly with the standard conception of model robustness as being governed by the bias-variance tradeoff. A simpler model, such as a linear SVM, can be expected to have higher bias (lower accuracy on clean data) but a lower variance, which allows it to generalize better as the test data becomes increasingly dissimilar to the training data as noise is added. However, their analysis is limited to low-dimensional data and binary classification problems, and no details about the specifications of the three SVM models are provided other than the kernel choice.

On the other hand, Ranzato and Zanella (2019) found the exact opposite trend when testing on high-dimensional image data, including MNIST. They identified the Radial Basis Function (RBF) kernel as the "most provably robust" option, whereas the linear kernel was shown to be very fragile to noise. The authors make no attempt to provide a theoretical explanation for their observed result.

However, the RBF kernel's observed superiority could be linked to the complex and localized geometry of its decision boundary. By effectively projecting the data into a higher-dimensional space and finding the optimal separating hyperplane there, the RBF kernel enables the model to define intricate decision surfaces that appear as complex, non-linear curves in the original input space. This flexibility allows the margin between classes to widen, positioning support vectors further from the decision boundary. Consequently, small perturbations to observations are less likely to shift them across the boundary, improving robustness to noise

However, it should be noted that this study employed a completely different methodology, using formal numerical verification to prove the stability of the various kernels in the face of inserted noise. Furthermore, they specifically focused on adversarial perturbations to the data, i.e. worst-case inputs intentionally crafted to deceive the classifier, rather than random, stochastic noise. Consequently, it remains unclear whether the RBF kernel's provable superiority against these targeted attacks generalizes to the unstructured noise more likely to be found in real-life scenarios.

3 Data & Method

3.1 Data

We base our experiments on the *Modified National Institute of Standards and Technology* (MNIST) dataset, a standard benchmark for handwritten digit classification consisting of 60,000 training and 10,000 test images. Figure 1 displays a random sample of ten images from this dataset.

Figure 1: Randomly selected MNIST digits illustrating the variability in handwriting.

Each observation $\mathbf{x}_i$ is a 28×28 pixel greyscale image of a centered handwritten digit, with corresponding label $y_i \in \{0, \ldots, 9\}$ (LeCun, Cortes, and Burges 2020).

Although the full dataset contains 60,000 training observations, we choose to restrict our training sample to 10,000 due to computational constraints and the superlinear time complexity of SVM training. While this may limit generalizability to some extent, the reduced sample remains sufficiently large to provide a meaningful basis for comparing kernel performance under noise.

This yields a dataset of 10,000 training observations and 10,000 test observations:

$$\mathbf{X}_{\text{train}} \in \mathbb{R}^{10{,}000 \times 784}, \qquad \mathbf{X}_{\text{test}} \in \mathbb{R}^{10{,}000 \times 784},$$

where each feature corresponds to a pixel intensity in $[0, 255]$, together with the label vectors:

$$\mathbf{y}_{\text{train}} \in \{0, \ldots, 9\}^{10{,}000}, \qquad \mathbf{y}_{\text{test}} \in \{0, \ldots, 9\}^{10{,}000}.$$

3.2 Limitations of the MNIST Dataset

Although MNIST remains a widely used benchmark for handwritten digit classification, it has several well-known shortcomings that should be acknowledged.

First, the dataset contains digits with distortions, irregular shapes, incomplete strokes, and substantial variation in skew across both the training and test sets (Amarnath and Vinay Kumar 2023). While this variability is part of what makes MNIST useful, it also reflects artifacts from the dataset construction process rather than naturally occurring handwriting noise.

Second, despite these distortions, the images are essentially free of background noise or blemishes, and lack realistic imperfections such as smudges or ink stains, as illustrated in Figure 1. This can cause models to perform well on MNIST while struggling with messier real-world inputs.

Finally, the dataset is highly curated and homogenized. As a result, reported test accuracy may provide an overly optimistic estimate of a model's true generalization ability,

since the controlled conditions of MNIST do not capture the full variability, noise, and irregularities present in handwritten digits encountered in production environments.

3.3 Model Specification & Parameter Tuning

We define three Support Vector Machine (SVM) models utilizing *linear*, *radial*, and *polynomial* kernels.[1]

1. **Linear SVM:** C
$$K(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^\top \mathbf{x}_j \tag{1}$$
2. **RBF SVM:** (C, γ)
$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2\right). \tag{2}$$
3. **Polynomial SVM:** (C, d)
$$K(\mathbf{x}, \mathbf{x}') = (\mathbf{x}^\top \mathbf{x}')^d \tag{3}$$

where

C : Regularization strength balancing margin size and classification error.
γ : Kernel coefficient determining the influence radius of individual samples.
d : Degree of the polynomial kernel controlling the model's nonlinearity.

To balance computational cost with sufficient exploration of the parameter space, we use compact grids that still capture meaningful variation in model complexity. The values for C, γ, and d span a reasonable range of regularization strength and nonlinearity. Before tuning, all features are standardized to have zero mean and unit variance. Based on this, we tune the hyperparameters using the following grids during 3-fold cross-validation:

- **Linear SVM:** $C \in \{0.1, 0.5, 1, 5\}$.
- **RBF SVM:** $C \in \{0.1, 0.5, 1, 5\}, \quad \gamma \in \{0.0001, 0.001, 0.01\}$.
- **Polynomial SVM:** $C \in \{0.1, 0.5, 1, 5\}, \quad d \in \{3, 4, 5\}, \quad \gamma = 1$.

The cross-validation procedure results in the following optimal hyperparameters:

Model	C	γ	d
Linear SVM	0.1	–	–
RBF SVM	5	0.01	–
Polynomial SVM	0.1	–	3

It is worth noting that the parameter C, which features in all three models and directly impacts generalizability by constraining misclassification, takes on a significantly higher value for the RBF model than for the two others. At first glance this may seem to compromise the comparability of the robustness results, but in the case of the RBF kernel, the C-parameter interacts with the γ-parameter (which regulates the complexity of the decision boundary), so the higher C value does not straightforwardly imply greater tolerance for misclassification. Additionally, initial testing showed that lower values for C dramatically reduced the performance of the RBF kernel to the extent that it could no longer serve as a meaningful baseline for comparison.

[1] The complete source code of the method is available in Byström and Stridh 2025.

3.4 Gaussian Noise Injection

To assess how well the trained Support Vector Machine (SVM) models handle degraded input data, we add Additive White Gaussian Noise (AWGN) to the normalized test matrix $\mathbf{X}_{\text{test}}$. This provides a simple way to mimic real-world imperfections such as sensor noise or poor image capture conditions.

3.4.1 Noise Model and Scaling

After normalizing the MNIST pixel values to the range $[0,1]$, we modify each feature $x_{i,j}$ by adding a noise term $\varepsilon_{i,j}$. The noise is sampled from a zero-mean normal distribution:

$$\varepsilon_{i,j} \sim \mathcal{N}(0, \sigma^2)$$

The noise level is controlled through the standard deviation σ, which we set directly using a percentage parameter p:

$$\sigma = p \quad \text{for } p \in \{0.0, 0.1, 0.2, \ldots, 1.0\}$$

This range allows us to examine how model performance changes as the noise increases from 0% to 100% of the full normalized intensity.

To illustrate the effect of increasing noise levels on individual digits, Figure 2 presents a randomly selected sample of three digits, $\{1,4,9\}$, each transformed according to Equation (4).

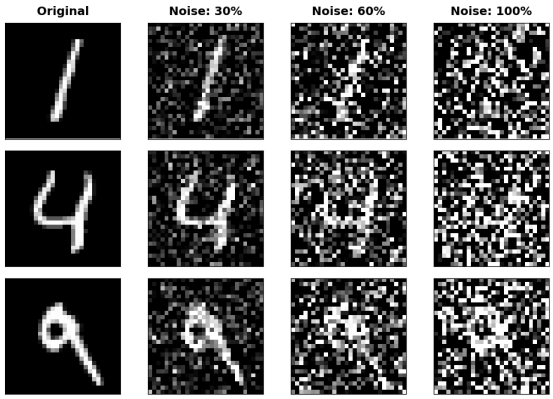

Figure 2: Examples of digits (1, 4, 9) under increasing Gaussian noise levels.

3.4.2 Adjustment and Clipping

The noisy features $\tilde{x}_{i,j}$ are obtained by adding the sampled noise to each normalized pixel value. Since valid pixel values lie in the interval $[0,1]$, we apply a clipping function to keep all adjusted values within this range:

$$\tilde{x}_{i,j} = \min(1, \max(0, x_{i,j} + \varepsilon_{i,j})) \tag{4}$$

For each noise level σ, we then compute the model's classification accuracy on the modified test set to evaluate how performance changes with increasing noise.

Since the noise is applied only to the test set, the models are always evaluated on inputs that differ from the clean training distribution. This provides a controlled way to assess robustness and compare how quickly performance degrades as the noise level increases.

4 Results

Table 1 displays the classification accuracies of all three SVM models in the uncorrupted MNIST data set. As can be expected in line with theory as well as previous work, the RBF model achieves the best classification accuracy on clean data, with the polynomial model not far behind. The linear model performs substantially worse due to the simple nature of its classification boundary, but it still achieves a remarkably high accuracy of 92.94%.

Kernel Type	**Accuracy** (%)
Linear	92.94
Polynomial	95.73
RBF	96.50

Table 1: Classification accuracies of all three models on clean MNIST data.

4.1 Error Analysis

Figure 3 presents confusion matrix heatmaps for all three SVM models on the uncorrupted MNIST dataset. Across all kernels, the diagonal elements dominate, indicating strong classification performance. The linear kernel displays the most pronounced off-diagonal values, consistent with its lower overall accuracy. The RBF and polynomial kernels yield cleaner confusion matrices, with misclassifications more sparsely distributed across digit pairs.

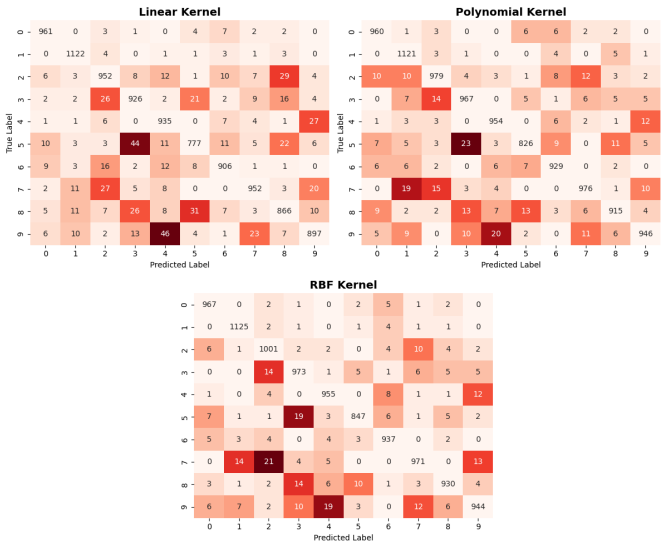

Figure 3: Confusion matrix heatmaps for all three models on clean MNIST data.

The three most common sources of misclassification for each model are displayed in Table 2. The patterns for all three models are similar, with $4 \leftrightarrow 9$ being the most common source of confusion for all three. $3 \leftrightarrow 5$ and $2 \leftrightarrow 7$ also feature prominently. This suggests that these digits tend to share similar features (pixel values), making it difficult for the SVM decision boundaries to separate them cleanly.

Model	Pair 1	Confusion	Pair 2	Confusion	Pair 3	Confusion
Linear	$4 \leftrightarrow 9$	73	$3 \leftrightarrow 5$	65	$5 \leftrightarrow 8$	53
RBF	$4 \leftrightarrow 9$	31	$2 \leftrightarrow 7$	31	$7 \leftrightarrow 9$	25
Polynomial	$4 \leftrightarrow 9$	32	$3 \leftrightarrow 5$	28	$2 \leftrightarrow 7$	27

Table 2: Top symmetrically confused pairs for all three models on clean MNIST data

Figure 4 corroborates this interpretation: when projecting the data into 2 dimensions using t-SNE, a dimensionality reduction technique that preserves local neighborhood structure, most digits form neat clusters with only small amounts of overlap. The pairs $4 \leftrightarrow 9$ and $3 \leftrightarrow 5$, however, occupy highly overlapping regions of space, and all of the other commonly confused pairs also demonstrate significant intermixing.

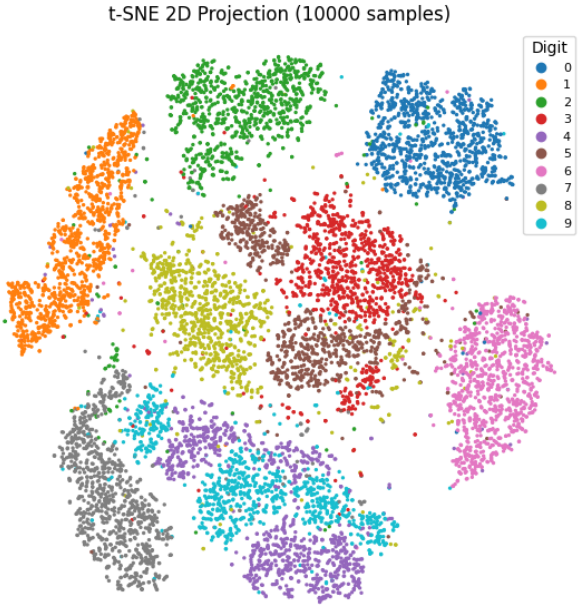

Figure 4: Two-dimensional t-SNE projection of the MNIST feature representations.

4.2 Robustness to Noise

As we add Gaussian noise to the images, it is natural to see a drop in accuracy. Since there are ten possible labels $y_i \in \{0, \ldots, 9\}$, a random guess would give an accuracy of about 10%.

From Table 3, we see that all three SVM models stay well above this baseline at every noise level. Even at 100% noise, where the digits are essentially impossible to recognize by eye, the models still perform noticeably better than random guessing. The polynomial SVM stands out in particular, since it keeps almost 40% accuracy at the highest noise level, which is much better than both the linear and RBF SVMs.

Noise (%)	Accuracy (%)		
	Linear SVM	**Polynomial SVM**	**RBF SVM**
0	92.94	95.73	96.50
10	92.50	95.57	96.12
20	89.64	94.50	94.87
30	79.86	91.51	87.63
40	65.71	84.59	73.86
50	54.09	74.13	55.19
60	44.59	64.76	38.20
70	37.97	55.42	30.00
80	33.40	49.30	24.61
90	30.28	44.20	21.86
100	27.35	39.95	19.07

Table 3: Accuracy of Linear, Polynomial, and RBF SVMs under increasing noise.

A possible reason for this difference in performance is the nature of the kernels. The polynomial kernel implicitly maps the data into a higher-degree feature space, which allows it to capture more flexible decision boundaries than the linear SVM. At the same time, it is less sensitive to small local distortions than the RBF kernel, which depends strongly on precise distance information. When the images become very noisy, this local structure breaks down, and the RBF model loses accuracy more quickly. The polynomial SVM can still rely on broader patterns in the transformed feature space, which may explain why it performs better as the noise level increases.

Another way to understand the behavior of the three models is through the complexity of their decision boundaries. The linear SVM is the simplest model, since it can only separate the data with a single linear hyperplane.[2] This gives it high bias and relatively low variance. Because of its simplicity, it performs reasonably well at low noise levels, but it cannot capture enough nonlinear structure to match the performance of the other kernels.

The polynomial SVM represents a middle ground. Its decision function is global, meaning that the shape of the boundary is influenced by the data as a whole, not only by nearby points. This global nature allows it to capture more structure than the linear SVM without becoming overly sensitive to small perturbations. As a result, it maintains better accuracy when noise increases and the fine local details of the images begin to disappear.

The RBF SVM is the most flexible model of the three. Its kernel is local, meaning that each support vector affects the decision boundary only in a small region around itself. This makes the model highly adaptive when the data is clean, but also more vulnerable when noise is added. As the images become noisy, the test points effectively ”move” in feature space, and the highly localized geometry of the RBF decision boundary makes it difficult for the model to generalize. This increased variance causes its accuracy to drop more sharply compared to the polynomial SVM.

[2]To clarify, for the linear case the hyperplane lives in the original feature space, which is not the case for non-linear kernels such as radial or polynomial.

Viewed from a bias-variance perspective, the results make sense: the linear SVM has high bias and low variance, the RBF SVM has low bias and high variance, and the polynomial SVM sits between them. Under heavy noise, a model with extremely high variance (such as the RBF SVM) tends to perform poorly, while one with extremely high bias (the linear SVM) cannot capture enough structure. The polynomial SVM provides the best compromise, which explains its consistently stronger performance as noise increases.

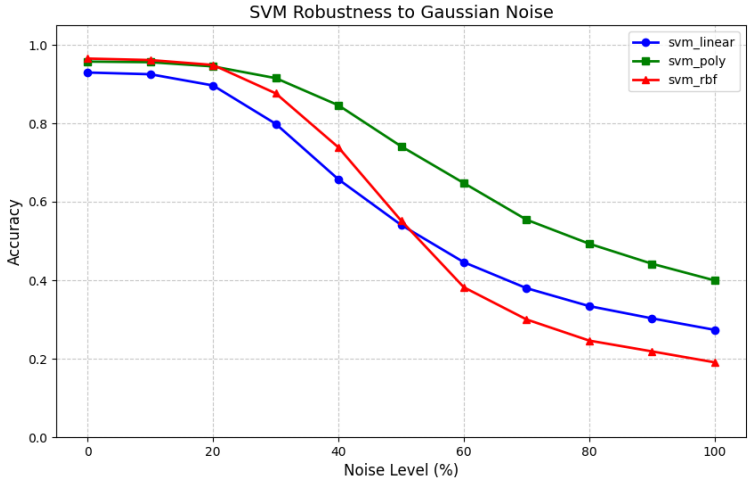

Figure 5: Accuracies of the three SVMs at different noise levels.

These results also help make sense of seemingly contradictory findings in the existing literature. Ljunggren and Ishii (2021) found that linear kernels were most robust to data corruption, Ranzato and Zanella (2019) identified the RBF kernel as most provably robust on MNIST. Our finding that the polynomial kernel outperforms both is not necessarily inconsistent with these earlier results once we account for differences in data dimensionality and noise type.

Ljunggren and Ishii tested on low-dimensional tabular datasets, the largest containing only 57 features. In such settings, a linear decision boundary is already reasonably expressive, so the variance reduction from using a simpler model outweighs the cost in bias. In 784 dimensions, however, there is enough structure that a linear boundary leaves substantial performance on the table, as shown by the roughly 3.5 percentage point gap on clean data (see table 1). This gap persists even at extreme noise levels: at 100% noise, the polynomial kernel still outperforms the linear by over 12 percentage points.

Ranzato and Zanella's finding can be reconciled by distinguishing between adversarial and stochastic noise. Adversarial perturbations are worst-case inputs specifically designed to cross the decision boundary with minimal displacement. The RBF kernel's localized boundaries can seemingly create protective margins against such targeted attacks, but this result does not seem to generalize to stochastic noise, which functions differently. When Gaussian noise perturbs each pixel independently, the polynomial kernel benefits from its global inner-product structure: zero-mean perturbations across 784 dimensions partially cancel out, which stabilizes kernel evaluations. The locality

that protects RBF against adversarial attacks becomes a liability when perturbations are random and unstructured.

5 Conclusion

This paper compared the robustness of linear, polynomial, and RBF support vector machines to Gaussian noise on MNIST. The polynomial kernel maintained the highest accuracy across most noise levels, outperforming both the simpler linear model and the more flexible RBF. This reflects its favorable position in the bias-variance tradeoff: it captures enough nonlinear structure to classify digits accurately while remaining stable as noise obscures fine local details.

These results fill a gap at the intersection of high-dimensional data and stochastic noise. Prior work established linear kernel dominance in low-dimensional settings and RBF robustness against adversarial attacks, but the behavior under random perturbations in high dimensions had not been systematically examined. Our findings suggest that kernel selection should account for both dimensionality and the expected noise regime.

Several limitations should be noted. We used a single configuration for each kernel, and different parameter choices or polynomial degrees might change the rankings. We also examined only attribute noise; label noise may produce different patterns. Future work could extend this analysis to other high-dimensional datasets and examine finer noise gradations below 10%.

References

Aizerman, Mark A., Emmanuel M. Braverman, and Lev I. Rozonoer (1964). "Theoretical foundations of the potential function method in pattern recognition learning". In: *Automation and Remote Control* 25.6, pp. 821–837.

Amarnath, R. and V. Vinay Kumar (2023). *Pruning Distorted Images in MNIST Handwritten Digits*. arXiv: 2307.14343 [cs.CV].

Boser, Bernhard E., Isabelle M. Guyon, and Vladimir Vapnik (1992). "A training algorithm for optimal margin classifiers". In: *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*. ACM, pp. 144–152.

Byström, Alexander Eriksson and Jack Andersson Stridh (2025). *stan52-project1-code (v1.0)*. Version master. Source code repository. URL: https://github.com/Alexerby/stan52-project1-code/releases/tag/v1.0 (visited on 11/26/2025).

Cortes, Corinna and Vladimir Vapnik (1995). "Support-Vector Networks". In: *Machine Learning* 20.3, pp. 273–297.

Cover, Thomas M. (1965). "Geometrical and Statistical Properties of Systems of Linear Inequalities with Applications in Pattern Recognition". In: *IEEE Transactions on Electronic Computers* EC-14.3, pp. 326–334.

LeCun, Yann, Corinna Cortes, and Christopher J. C. Burges (2020). *The MNIST Database of Handwritten Digits*. Accessed 30 April 2020. URL: https://web.archive.org/web/20200430193701/http://yann.lecun.com/exdb/mnist/.

Ljunggren, Fredrik and Yuta Ishii (2021). "Robustness of Support Vector Machines: A study on the robustness of different kernels against noise". Bachelor's Thesis. KTH Royal Institute of Technology.

Ranzato, Francesco and Marco Zanella (2019). "Robustness Verification of Support Vector Machines". In: *Proceedings of the 23rd International Symposium on Formal Methods (FM 2019)*. Vol. 11800. Lecture Notes in Computer Science. Springer, pp. 271–288.

Vapnik, Vladimir and Alexey Chervonenkis (1964). "A note on one class of perceptrons". In: *Automation and Remote Control* 25.1. Translated from Avtomatika i Telemekhanika, pp. 103–109.