RISK AVERSE OPTIMIZATION

RESEARCH CASE STUDY REPORT

for the degree of
Master of Data Science

presented at the Department IV
of Trier University

submitted by

Rahul, Krishnan, Universitätsring 8E, 54296 Trier

Supervisor: Prof. Dr. Volker Schulz

Trier, February 6, 2026

UNIVERSITÄT TRIER

Eigenständigkeitserklärung

Hiermit versichere ich,

Name, Vorname _____

Matrikelnummer _____

dass ich die vorliegende schriftliche Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt und die aus fremden Quellen direkt oder indirekt übernommenen Gedanken und Inhalte als solche kenntlich gemacht habe. Die beigefügte Arbeit habe ich bisher nicht zum Erwerb eines anderen Leistungsnachweises eingereicht und keinem anderen Prüfungsamt in gleicher oder vergleichbarer Form vorgelegt. Sie wurde bisher nicht veröffentlicht.

Ich versichere, dass ich auf generativer Künstlicher Intelligenz (genKI) basierende text- oder sonstige inhaltgenerierende Hilfsmittel nur auf die durch die Prüferin/den Prüfer schriftlich gestattete Weise verwendet und genKI generierte Passagen explizit gekennzeichnet habe.
Bei Verwendung von genKI-Tools habe ich diese mit ihrem Produktnamen angegeben und in einer Übersicht vollständig aufgeführt.

Ich verantworte die Übernahme der von mir verwendeten genKI generierten Passagen und Inhalte in meiner Arbeit vollumfänglich selbst.

Titel der Arbeit _____

Studienfach _____

Prüferin/Prüfer _____

Es handelt sich um eine Leistung für eine

- ☐ Bachelorarbeit
- ☐ Masterarbeit
- ☐ sonstige Prüfungsleistung für Modul (Hausarbeit etc.) _____

_____ _____

Ort, Datum Unterschrift

Stand: 06/2025

Contents

1 Introduction

Risk-averse optimization concerns decision problems under uncertainty in which rare but adverse outcomes must be controlled explicitly. In contrast to risk-neutral formulations that minimize expected values, risk-averse approaches account for tail events that can dominate overall performance. This distinction is particularly relevant in numerical optimization and optimal control, where infrequent unfavorable scenarios may lead to severe degradation or failure of the system [8].

A prominent risk measure for capturing tail behavior is the Conditional Value-at-Risk (CVaR). Originally developed in financial mathematics, CVaR is now a standard risk measure and is characterized by a variational formulation due to Rockafellar and Uryasev. This representation expresses CVaR as the solution of an auxiliary optimization problem and establishes its convexity with respect to additional variables, while the full problem may remain nonconvex in the original decision variables [6, 7]. All risk-averse formulations in this report are based on this approach.

Although CVaR is well established in finance, its systematic use in nonlinear optimization and optimal control has gained broader attention only in recent years. From a numerical perspective, CVaR-based problems exhibit a structure that is well suited to scenario-based approximations and modern optimization algorithms, as studied in stochastic programming and numerical optimization literature [8, 4].

This report investigates CVaR-based optimization through three case studies of increasing complexity, motivated by the T2 Mathematics module on risk-averse optimization. The first case study considers a risk-averse wobbly Rosenbrock problem as a minimal example under parametric uncertainty. The second case study addresses a risk-averse support vector machine, illustrating CVaR in a high-dimensional convex learning problem with noisy data [10]. The third and most involved case study focuses on risk-averse model predictive control for wine fermentation, where uncertainty in ambient temperature can lead to severe process failures if tail risks are ignored [5].

The objective of this work is to study the numerical implementation of CVaR-based formulations using scenario approximations and contemporary optimization software, and to assess their behavior in comparison to risk-neutral approaches. Emphasis is placed on reproducibility, solver performance, and the trade-off between robustness and computational effort.

2 Mathematical and Computational Fundamentals

This chapter introduces the mathematical concepts and numerical tools required for the formulation and solution of the risk-averse optimization problems considered in this report. The presentation is intentionally restricted to material that is used directly in the subsequent case studies, in accordance with standard guidelines for scientific theses in numerical mathematics.

2.1 Risk Measures in Optimization

Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space and let $L : X \times \Omega \to \mathbb{R}$ denote a loss function depending on a decision variable $x \in X \subset \mathbb{R}^n$ and a random variable $\xi \in \Omega$.

The classical risk-neutral optimization problem minimizes the expected loss

$$\min_{x \in X} \mathbb{E}\big[L(x, \xi)\big]. \tag{2.1}$$

While this formulation is mathematically convenient, it does not distinguish between moderate and extreme losses. Rare but severe events may therefore be underrepresented, which can lead to solutions with poor robustness properties in practice [8].

To address this limitation, risk measures are introduced. A risk measure is a functional

$$\rho : L^p(\Omega) \to \mathbb{R} \tag{2.2}$$

that assigns a scalar risk value to a random loss. Important examples include the expectation, the variance, the Value-at-Risk (VaR), and the Conditional Value-at-Risk (CVaR). Among these, CVaR plays a central role in this work due to its favorable analytical and numerical properties.

2.2 Value-at-Risk and Conditional Value-at-Risk

For a given confidence level $\alpha \in (0, 1)$, the Value-at-Risk of a loss random variable Z is defined as

$$\mathrm{VaR}_\alpha(Z) := \inf\big\{ t \in \mathbb{R} \,\big|\, \mathbb{P}(Z \le t) \ge \alpha \big\}. \tag{2.3}$$

VaR characterizes a quantile of the loss distribution but provides no information about the magnitude of losses beyond this threshold. Moreover, VaR is in general nonconvex and noncoherent, which limits its usefulness in optimization problems.

The Conditional Value-at-Risk (CVaR) remedies these shortcomings by averaging losses in the tail of the distribution:

$$\mathrm{CVaR}_\alpha(Z) := \mathbb{E}\big[Z \,\big|\, Z \geq \mathrm{VaR}_\alpha(Z)\big]. \tag{2.4}$$

CVaR is a coherent risk measure and explicitly penalizes extreme outcomes. Its key advantage for optimization lies in the variational representation introduced by Rockafellar and Uryasev [6, 7]:

$$\mathrm{CVaR}_\alpha(Z) = \min_{t\in\mathbb{R}} \left\{ t + \frac{1}{1-\alpha}\,\mathbb{E}\big[(Z-t)_+\big] \right\}, \tag{2.5}$$

where $(\cdot)_+ := \max\{0, \cdot\}$. This formulation transforms CVaR minimization into a standard optimization problem with auxiliary variables. The resulting objective is convex in the risk-related variables, even if the original loss function is nonconvex in the decision variable x. All CVaR-based formulations in this report rely on this representation.

2.3 Scenario Approximation (Sample Average Approximation)

In practical applications, the distribution of ξ is rarely available in closed form. Expectations are therefore approximated using Monte Carlo sampling. Given N independent samples $\{\xi^{(i)}\}_{i=1}^N$, the expectation term in the CVaR formulation is replaced by a sample average, yielding the Sample Average Approximation (SAA):

$$\min_{x,t} \; t + \frac{1}{(1-\alpha)N} \sum_{i=1}^{N} \big(L(x,\xi^{(i)}) - t\big)_+. \tag{2.6}$$

Introducing auxiliary variables $u_i \geq 0$ leads to an equivalent finite-dimensional constrained optimization problem:

$$\min_{x,t,u} \; t + \frac{1}{(1-\alpha)N} \sum_{i=1}^{N} u_i, \tag{2.7}$$

$$\text{s.t.} \quad u_i \geq L(x,\xi^{(i)}) - t, \quad i = 1,\dots,N, \tag{2.8}$$

$$u_i \geq 0, \quad i = 1,\dots,N, \tag{2.9}$$

$$x \in X. \tag{2.10}$$

Under mild assumptions, solutions of the SAA problem converge to solutions of the original stochastic problem as $N \to \infty$ [8]. This approach forms the numerical backbone of all case studies considered in this work.

2.4 Numerical Optimization and Software Tools

The optimization problems arising from CVaR formulations take the form of nonlinear programs (NLPs) or quadratic programs (QPs), depending on the structure of the underlying loss function. In the static optimization and optimal control case studies, NLPs are solved using interior-point methods, while convex learning problems lead to large-scale QPs [4].

For nonlinear problems, the implementations rely on CasADi, which provides symbolic expressions, automatic differentiation, and seamless interfaces to NLP solvers such as IPOPT [1, 11]. Convex quadratic problems are formulated using CVXPY [2] and solved with operator-splitting methods.

The numerical behavior of these solvers, including robustness, convergence properties, and computational cost, is an integral part of the analysis in the subsequent chapters.

3 Case Study I: Risk-Averse Wobbly Rosenbrock Problem

3.1 Problem Formulation and Uncertainty Model

As a minimal yet illustrative example of risk-averse optimization, we consider a perturbed version of the classical Rosenbrock function [**?**]. The deterministic Rosenbrock function with parameters $a \in \mathbb{R}$ and $b > 0$ is defined as

$$\phi(x; a, b) = (a - x_1)^2 + b\,(x_2 - x_1^2)^2, \qquad x = (x_1, x_2)^\top. \tag{3.1}$$

For the standard choice $a = 1$ and $b = 100$, the function has a unique global minimizer at $x^\star = (1, 1)$.

Uncertainty is introduced through additive noise in the parameter a,

$$a(\xi_a) = a + \xi_a, \tag{3.2}$$

which leads to the random loss

$$f(x, \xi_a) = \phi\big(x; a + \xi_a, b\big). \tag{3.3}$$

This setting models parametric uncertainty, where the decision variable itself is deterministic but the system parameters are subject to random perturbations.

The decision variable is constrained to the compact feasible set

$$X = [-2, 2]^2, \tag{3.4}$$

ensuring boundedness of the optimization problem.

3.2 CVaR-Based Optimization

For a confidence level $\alpha \in (0, 1)$, the objective is to minimize the Conditional Value-at-Risk of the random loss,

$$\min_{x \in X} \ \mathrm{CVaR}_\alpha\big[f(x, \xi_a)\big]. \tag{3.5}$$

Using the Rockafellar–Uryasev variational representation, CVaR can be written as [6, 7]

$$\mathrm{CVaR}_\alpha[f] = \min_{t\in\mathbb{R}} \left\{ t + \frac{1}{1-\alpha}\,\mathbb{E}\big[(f-t)_+\big] \right\}, \tag{3.6}$$

where t represents the Value-at-Risk threshold and $(\cdot)_+ = \max\{0, \cdot\}$. This formulation transforms the risk-averse problem into a standard constrained optimization problem that is convex in the auxiliary variables but generally nonconvex in x.

Since the expectation cannot be evaluated analytically, a Sample Average Approximation (SAA) is employed [8]. Given N independent samples $\{\xi_a^{(i)}\}_{i=1}^N$, the finite-dimensional problem becomes

$$\min_{x,t,u}\ t + \frac{1}{(1-\alpha)N}\sum_{i=1}^{N} u_i, \tag{3.7}$$

$$\text{s.t.}\quad u_i \geq \phi\big(x; a+\xi_a^{(i)}, b\big) - t, \quad i = 1,\dots,N, \tag{3.8}$$

$$u_i \geq 0, \quad i = 1,\dots,N, \tag{3.9}$$

$$x \in X. \tag{3.10}$$

Despite the nonconvexity in x, the problem is small-scale and can be solved reliably using standard NLP solvers (here via CasADi/IPOPT) [1, 11].

3.3 Numerical Experiment Setup

All experiments in this chapter are conducted with $b = 100$ and a CVaR confidence level of $\alpha = 0.99$, placing strong emphasis on tail risk. The CVaR optimization problem is solved using $N = 500$ training scenarios generated via SAA. The resulting solutions are evaluated out-of-sample using $M = 5000$ independent realizations of the parameter noise.

Two different noise models for the parameter perturbation ξ_a are considered:

- Standard normal distributed noise, representing light-tailed uncertainty.
- Student's t-distributed noise, representing heavy-tailed uncertainty with rare extreme events.

For each noise model, two solutions are compared:

- a nominal solution, obtained by minimizing the deterministic Rosenbrock function,
- a CVaR-optimized solution, obtained from the SAA-based CVaR formulation.

Performance is assessed using the out-of-sample mean loss and the empirical $\mathrm{CVaR}_{0.99}$, computed as the mean of the worst 1% of observed losses.

3.4 Numerical Results and Discussion

The numerical results of the wobbly Rosenbrock experiment are summarized in Table 3.1, which reports out-of-sample performance for both the nominal and CVaR-optimized solutions under standard normal distributed and Student's t-distributed parameter noise. Performance is evaluated using the empirical mean loss and the empirical $\mathrm{CVaR}_{0.99}$, computed from $M = 5000$ independent test samples.

Table 3.1: Out-of-sample performance for the wobbly Rosenbrock problem ($\alpha = 0.99$, $N = 500$, $M = 5000$).

Noise distribution	Solution	x_1	x_2	Mean loss	$\mathrm{CVaR}_{0.99}$
Standard normal distributed	Nominal	1.000	1.000	0.002417	0.020196
Standard normal distributed	CVaR	0.997	0.993	0.002417	0.019941
Student's t	Nominal	1.000	1.000	0.007048	0.237490
Student's t	CVaR	0.950	0.903	0.009584	0.234932

Under standard normal distributed parameter noise, the CVaR-optimized solution exhibits essentially the same mean loss as the nominal solution, while achieving a small but consistent reduction in $\mathrm{CVaR}_{0.99}$. As shown in Table 3.1, the CVaR value decreases from 0.020196 for the nominal solution to 0.019941 for the CVaR-optimized solution, without any noticeable degradation in average performance. This indicates that, for light-tailed uncertainty, the deterministic minimizer of the Rosenbrock function is already close to optimal, and CVaR optimization provides only marginal improvements in tail risk.

The effect of CVaR optimization becomes more pronounced under heavy-tailed Student's t-distributed noise. In this case, the nominal solution exhibits larger tail losses, resulting in a $\mathrm{CVaR}_{0.99}$ of 0.237490. The CVaR-optimized solution reduces this value to 0.234932, at the cost of an increased mean loss. This behavior reflects the defining characteristic of risk-averse optimization: the solution is adjusted to suppress rare but severe outcomes, even when this leads to inferior average performance. The corresponding shift of the optimizer away from the deterministic minimizer illustrates how CVaR explicitly responds to heavy-tailed uncertainty by prioritizing tail behavior over typical realizations.

Taken together, these results confirm the central insight of this case study. Risk-averse optimization via CVaR does not universally outperform risk-neutral optimization. When uncertainty is light-tailed and the nominal optimum is inherently robust, CVaR yields only minor gains. In contrast, when uncertainty exhibits heavy tails and extreme events dominate the loss distribution, CVaR provides a clear and systematic mechanism for controlling tail risk, even at the expense of increased mean loss. This observation motivates the subsequent case studies, where uncertainty and catastrophic failure modes play a more decisive role.

4 Case Study II: Risk-Averse Support Vector Machine

4.1 Problem Description and Motivation

This case study investigates risk-averse optimization in a high-dimensional machine learning setting. Unlike the Rosenbrock example, where uncertainty affects a low-dimensional parametric objective, here uncertainty enters through perturbations of the input data itself. Such perturbations can lead to localized but severe degradation in classification performance, particularly for image-based tasks.

Support Vector Machines (SVMs) are a standard method for binary classification and are typically trained by minimizing the average hinge loss [10]. This risk-neutral approach focuses on mean performance and does not explicitly control the impact of rare but unfavorable perturbations. The objective of this case study is to examine whether minimizing the Conditional Value-at-Risk (CVaR) of the hinge loss using the Rockafellar–Uryasev formulation [6, 7] leads to classifiers that are more robust to noisy inputs.

4.2 Nominal SVM Formulation

For a labeled dataset $\{(x_i, y_i)\}_{i=1}^n$, the standard linear SVM is commonly introduced by minimizing the hinge loss

$$\max\big(0,\, 1 - y_i(w^\top x_i - b)\big), \tag{4.1}$$

leading to the risk-neutral optimization problem

$$\min_{w,b}\ \frac{1}{n}\sum_{i=1}^{n} \max\big(0,\, 1 - y_i(w^\top x_i - b)\big) + \lambda\|w\|_2^2. \tag{4.2}$$

The hinge loss is convex but non-differentiable due to the pointwise maximum. In order to obtain a formulation suitable for quadratic programming solvers, this objective is rewritten

using slack variables ζ_i, yielding the equivalent primal soft-margin SVM formulation [10]:

$$\min_{w,b,\zeta} \frac{1}{n}\sum_{i=1}^{n}\zeta_i + \lambda\|w\|_2^2, \tag{4.3}$$

$$\text{s.t.} \quad y_i(w^\top x_i - b) \geq 1 - \zeta_i, \quad i = 1, \ldots, n, \tag{4.4}$$

$$\zeta_i \geq 0, \quad i = 1, \ldots, n. \tag{4.5}$$

This reformulation replaces the non-smooth max operator with linear constraints and auxiliary variables, while preserving convexity. The resulting problem has a quadratic objective and linear constraints and therefore constitutes a convex quadratic program (QP). This is the formulation used in the numerical implementation.

4.3 CVaR-Based SVM Formulation

To incorporate robustness against input perturbations, the SVM training objective is extended from a risk-neutral to a risk-averse formulation using the Conditional Value-at-Risk (CVaR) [6, 7]. Uncertainty is modeled through additive perturbations of the input data. For each original training sample x_i, multiple perturbed versions are generated,

$$x_i^{(k)} = x_i + \delta_i^{(k)}, \qquad k = 1, \ldots, N, \tag{4.6}$$

where each index k corresponds to one perturbation scenario of the entire dataset.

For a fixed classifier (w, b), the hinge loss in scenario k is computed using the slack-variable formulation introduced in Section 4.2. Let $\zeta_i^{(k)}$ denote the slack variable associated with sample i in scenario k. The average hinge loss for scenario k is then given by

$$\ell^{(k)}(w, b) = \frac{1}{n}\sum_{i=1}^{n}\zeta_i^{(k)}. \tag{4.7}$$

The risk-averse training objective minimizes the CVaR of the scenario-dependent loss distribution $\{\ell^{(k)}\}_{k=1}^{N}$. Using the Rockafellar–Uryasev representation, CVaR can be written as

$$\mathrm{CVaR}_\alpha[\ell] = \min_{t\in\mathbb{R}} \left\{ t + \frac{1}{1-\alpha}\,\mathbb{E}\big[(\ell - t)_+\big] \right\}, \tag{4.8}$$

where $\alpha \in (0, 1)$ is the confidence level and t represents the Value-at-Risk threshold.

Approximating the expectation by a sample average over N perturbation scenarios (SAA)

[8] yields the finite-dimensional optimization problem

$$\min_{w,b,\zeta,t,u} \lambda\|w\|_2^2 + t + \frac{1}{(1-\alpha)N}\sum_{k=1}^{N} u_k, \tag{4.9}$$

$$\text{s.t.} \quad y_i\big(w^\top x_i^{(k)} - b\big) \geq 1 - \zeta_i^{(k)}, \quad \forall i,\ k, \tag{4.10}$$

$$\zeta_i^{(k)} \geq 0, \quad \forall i,\ k, \tag{4.11}$$

$$u_k \geq \ell^{(k)}(w,b) - t, \quad k = 1,\ldots,N, \tag{4.12}$$

$$u_k \geq 0, \quad k = 1,\ldots,N. \tag{4.13}$$

This formulation introduces auxiliary variables $\zeta_i^{(k)}$ to handle the non-smooth hinge loss and variables t and u_k to aggregate losses in the upper tail of the distribution. Importantly, all constraints are linear, and the objective function consists of a quadratic regularization term and linear terms in the auxiliary variables. As a result, the CVaR-based SVM formulation remains a convex quadratic program (QP) [10].

In contrast to the Rosenbrock case study, where nonconvexity arises from the objective function itself, both the nominal and CVaR-based SVM problems are convex and can be solved efficiently using standard QP solvers. This convex structure allows robustness against input perturbations to be achieved without sacrificing computational tractability.

4.4 Numerical Experiment Setup

This section describes the experimental configuration used to evaluate the nominal and CVaR-optimized SVM classifiers. The objective is to ensure transparency, reproducibility, and a fair comparison between risk-neutral and risk-averse training.

4.4.1 Dataset and Preprocessing

The experiments are conducted on a binary image classification task (cats versus dogs) [3]. Due to system and computational constraints, the dataset is restricted to a balanced subset of 100 images in total, consisting of an equal number of cat and dog images. This subset is sufficient to demonstrate the effect of risk-averse training while keeping the computational cost of scenario-based CVaR optimization manageable.

All images are resized to a fixed resolution, converted to grayscale, flattened into feature vectors, and normalized. The dataset is split into disjoint training and test sets of equal size.

4.4.2 Perturbation Model and Reproducibility

Input uncertainty is introduced through additive noise applied to the image pixels. Two perturbation models are considered:

- Standard normal perturbations, representing light-tailed uncertainty.
- Student's t-distributed perturbations with df=2, representing heavy-tailed uncertainty.

Perturbations are applied independently to each pixel. For both training and evaluation, the same perturbation realizations are used for the nominal and CVaR-based classifiers. This ensures that any observed performance differences are attributable solely to the training objective and not to differences in noise realizations.

All perturbations are generated using a fixed random seed. This guarantees full reproducibility of the experiments and allows for a direct, controlled comparison between the nominal and CVaR-optimized solutions.

4.4.3 Nominal Training Configuration

The nominal SVM is trained using the primal soft-margin formulation described in Section 4.2. The regularization parameter λ is fixed throughout all experiments. Training is performed on the unperturbed training data, corresponding to a standard risk-neutral classifier.

4.4.4 CVaR Training Configuration

For CVaR-based training, robustness is enforced at the level of the loss distribution rather than at individual samples. The CVaR confidence level is fixed to $\alpha = 0.95$, corresponding to optimization with respect to the worst 5% of perturbation scenarios.

During training, $N =$ independent perturbation scenarios of the entire training dataset are generated. For each scenario, the average hinge loss is computed using the slack-variable formulation. These scenario losses are aggregated using the CVaR objective, resulting in the convex quadratic program described in Section 4.3 [6, 7, 8].

4.4.5 Solver and Implementation Details

All optimization problems in this case study are implemented in CVXPY [2] and solved using the OSQP solver [9]. The nominal SVM and the CVaR-based SVM are both formulated as convex quadratic programs (QPs), consisting of a quadratic objective and linear constraints, as described in Sections 4.2 and 4.3. Solver tolerances are explicitly

specified, with absolute and relative convergence tolerances set to 10^{-5}, and a maximum iteration limit of 10 000 to ensure convergence in all runs.

4.4.6 Evaluation

Classifier performance is evaluated using three complementary metrics:

1. Unperturbed test accuracy, measured on clean test data.
2. Mean perturbed test accuracy, averaged over $M = 100$ independently perturbed test sets.
3. $\text{CVaR}_{0.95}$ perturbed test accuracy, capturing worst-case performance under severe perturbations.

In addition, the mean hinge loss and $\text{CVaR}_{0.95}$ hinge loss are computed over the perturbed test sets to directly assess tail behavior in the loss distribution.

4.5 Numerical Results and Discussion

This section presents and interprets the numerical performance of the nominal and CVaR-optimized SVM classifiers under both light-tailed and heavy-tailed input perturbations. All experiments are conducted using the configuration: Dimension = 64, $\lambda = 10^{-3}$, $\alpha = 0.95$, $N = 10$, $M = 100$, and $\sigma = 0.8$, with grayscale images and a balanced dataset consisting of 100 cat and 100 dog images [3]. Identical perturbation realizations are used for both classifiers to ensure a fair comparison.

Table 4.1: SVM performance under input perturbations ($\alpha = 0.95$, $N = 10$, $M = 100$).

Perturbation type	Model	Mean hinge loss	$\text{CVaR}_{0.95}$ loss	Acc. (unpert.)	Acc. (mean)	Acc. ($\text{CVaR}_{0.95}$)
Std. normal	Nominal	1.048	1.115	53.0%	51.89%	47.29%
Std. normal	CVaR SVM	**0.978**	**1.029**	**58.0%**	**54.53%**	**49.29%**
Student's t	Nominal	1.064	1.160	53.0%	51.49%	45.71%
Student's t	CVaR SVM	**0.958**	**1.033**	**57.0%**	**54.86%**	**48.22%**

Table 4.1 provides a detailed comparison between the nominal and CVaR-optimized SVM classifiers under both standard normal and Student's t-distributed input perturbations. The reported metrics allow performance to be assessed not only in terms of average behavior but also with respect to tail risk, which is the primary focus of CVaR-based optimization.

Under standard normal perturbations, the nominal SVM exhibits a noticeable gap between mean perturbed accuracy (51.89%) and $\text{CVaR}_{0.95}$ accuracy (47.29%), indicating sensitivity to unfavorable noise realizations. This behavior is also reflected in the hinge loss statistics: while the mean hinge loss remains moderate, the $\text{CVaR}_{0.95}$ hinge loss increases, revealing the presence of high-loss tail events. The CVaR-optimized SVM improves performance across all reported metrics in this regime. As shown in Table 4.1, both the mean and $\text{CVaR}_{0.95}$ hinge losses are reduced, and classification accuracy increases not only in the worst-case tail but also on average and on the unperturbed test set. This suggests that CVaR training suppresses extreme-loss scenarios without sacrificing baseline performance. In fact, the improvement in unperturbed accuracy indicates that the risk-neutral classifier may overfit noise-sensitive directions in the feature space, while the CVaR objective acts as an implicit regularizer. Some sample images where the CVaR svm classifies correctly while the nominal svm misclassifies is shown in Figure 4.1.

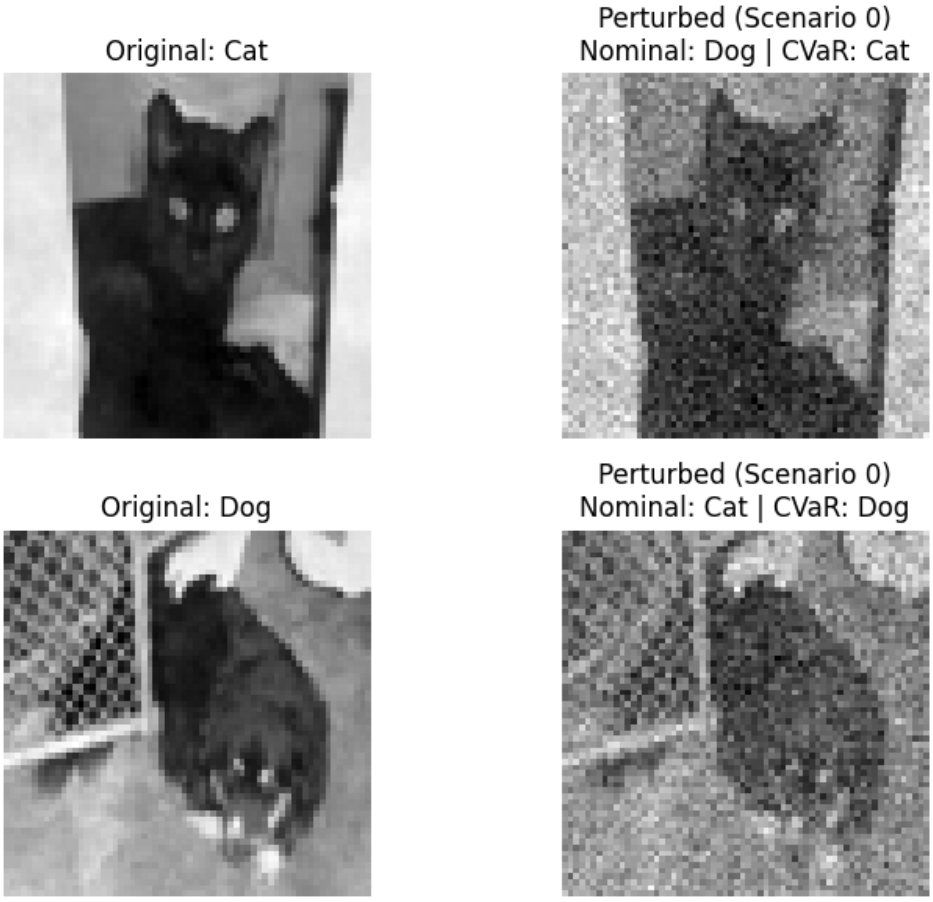

Figure 4.1: 2 sample images from the test set where nominal solution misclassified and CVaR solution classified correctly

The advantages of CVaR-based training become more pronounced under Student's t-distributed perturbations, which introduce heavy-tailed noise and rare but severe distor-

tions. In this setting, the nominal SVM shows a substantial degradation in tail performance, with a $\text{CVaR}_{0.95}$ hinge loss of 1.160 and a $\text{CVaR}_{0.95}$ accuracy of only 45.71%, despite a mean perturbed accuracy close to 51%. This discrepancy highlights that extreme perturbations dominate worst-case behavior even when average performance appears acceptable. The CVaR-optimized SVM significantly mitigates this effect. As reported in Table 4.1, both the mean and $\text{CVaR}_{0.95}$ hinge losses are reduced, and the $\text{CVaR}_{0.95}$ accuracy improves by more than two percentage points. Importantly, these robustness gains are not achieved at the expense of overall predictive quality, as unperturbed and mean perturbed accuracies also increase.

Across both perturbation models, a consistent pattern emerges. The nominal SVM is vulnerable to tail events, as evidenced by the systematic drop from mean perturbed accuracy to $\text{CVaR}_{0.95}$ accuracy. In contrast, the CVaR-optimized SVM reduces tail hinge loss and improves worst-case accuracy in all cases considered. The magnitude of this improvement is modest under light-tailed noise but becomes clearly visible under heavy-tailed perturbations, which aligns with the theoretical motivation of CVaR. These results contrast with the Rosenbrock case study, where the nominal solution was already relatively robust under light-tailed uncertainty. In the high-dimensional SVM setting, localized pixel-level perturbations can strongly influence classifier decisions, making tail-aware optimization substantially more effective.

Overall, the results summarized in Table 4.1 demonstrate that CVaR-based SVM training provides a meaningful robustness advantage in the presence of input noise, particularly when uncertainty exhibits heavy tails, while preserving or even improving average-case performance.

Bibliography

[1] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl. Casadi – a software framework for nonlinear optimization and optimal control. *Mathematical Programming Computation*, 2019.

[2] S. Diamond and S. Boyd. Cvxpy: A python-embedded modeling language for convex optimization. *Journal of Machine Learning Research*, 2016.

[3] Kaggle. Dogs vs. cats. Dataset. URL: https://www.kaggle.com/competitions/dogs-vs-cats/data.

[4] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer.

[5] J. B. Rawlings and D. Q. Mayne. *Model Predictive Control: Theory and Design*.

[6] R. T. Rockafellar and S. Uryasev. Optimization of conditional value-at-risk. *Journal of Risk*, 2000.

[7] R. T. Rockafellar and S. Uryasev. Conditional value-at-risk for general loss distributions. *Journal of Banking & Finance*, 2002.

[8] A. Shapiro, D. Dentcheva, and A. Ruszczyński. *Lectures on Stochastic Programming*. SIAM.

[9] B. Stellato, G. Banjac, P. Goulart, A. Bemporad, and S. Boyd. OSQP: An operator splitting solver for quadratic programs. *Mathematical Programming Computation*, 2020.

[10] V. N. Vapnik. *Statistical Learning Theory*. Wiley.

[11] A. Wächter and L. T. Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 2006.