WROCŁAW UNIVERSITY OF SCIENCE AND TECHNOLOGY
FACULTY OF ELECTRONICS

Doctoral Dissertation

Rigorous Evaluation: Towards Trustworthy Representations in Machine Learning

Rzetelna ewaluacja: w stronę wiarygodnych reprezentacji w uczeniu maszynowym

AUTHOR:
Mateusz Gniewkowski, MSc

SUPERVISOR:
Henryk Maciejewski, D.Sc
Tomasz Surmacz, PhD

WROCŁAW 2025

Acknowledgements

First and foremost, I would like to express my sincere gratitude to my supervisors, Henryk Maciejewski, D.Sc., and Tomasz Surmacz, Ph.D., for their invaluable support, numerous discussions, and guidance at every stage of this dissertation. I have no doubt that without their commitment and patience, this work would never have been completed.

I am also deeply grateful to our academic community – to the people who gave me the opportunity to grow, who created a welcoming atmosphere, and who worked side by side with me. In particular, I would like to thank Piotr Bielak, Szymon Datko, Marek Klonowski, Kamil Szyc, Paweł Walkowiak, and Tomasz Walkowiak, as well as many others who cannot all be named here.

I owe my deepest thanks to my family - my mother, Marta Gniewkowska, and my grandmother, Wiesława Gniewkowska - as well as to all my friends. Thank you for your patience and support, as I am fully aware that this work has taken a toll on my social life.

Finally, I would like to thank everyone who will read this dissertation. I hope that this work – despite its limitations – will prove to be interesting, inspiring, and thought-provoking.

Mateusz Gniewkowski

Podziękowania

Przede wszystkim chciałbym serdecznie podziękować moim promotorom, Profesorowi Henrykowi Maciejewskiemu oraz Doktorowi Tomaszowi Surmaczowi, za nieocenione wsparcie, liczne dyskusje i pomoc na każdym etapie tej pracy. Nie mam wątpliwości, że bez Waszego zaangażowania i cierpliwości ta rozprawa nigdy by nie powstała.

Jestem również niezwykle wdzięczny całej naszej społeczności akademickiej – ludziom, którzy dali mi szansę na rozwój, stworzyli przyjazną atmosferę i pracowali ze mną ramię w ramię. W szczególności dziękuję: Piotrowi Bielakowi, Szymonowi Datce, Markowi Klonowskiemu, Kamilowi Szycowi, Pawłowi Walkowiakowi oraz Tomaszowi Walkowiakowi, a także zapewne wielu innym osobom, których nie sposób wszystkich tutaj wymienić.

Dziękuję swojej rodzinie – mamie, Marcie Gniewkowskiej, oraz babci, Wiesławie Gniewkowskiej – a także wszystkim moim znajomym i przyjaciołom. Jestem ogromnie wdzięczny za wsparcie i cierpliwość, bo nie mam wątpliwości, że ta praca kosztowała mnie sporo życia towarzyskiego.

Na końcu pragnę podziękować wszystkim, którzy zechcą przeczytać tę rozprawę. Mam nadzieję, że ta praca – mimo swoich ograniczeń – będzie stanowiła lekturę ciekawą, inspirującą i pouczającą.

Mateusz Gniewkowski

Abstract

The starting point of this dissertation is the gap between "single-number" evaluation and the actual reliability of models, that is, their behavior in real-world conditions. High accuracy on data drawn from the same distribution as the training set (*Independent and Identically Distributed*, IID) does not tell us whether the model bases its decision on appropriate evidence, nor whether its representations are useful beyond a narrow task. We also do not know how the model will behave under a distribution shift or under small, semantically insignificant changes. In practice, real-world systems encounter environments full of correlations, novelties, and stimuli that can easily "fool" a classifier. The gap between test and deployment arises in particular from: nonstationarity and domain shift; narrow, nonrepresentative validation procedures (no out-of-distribution assessment and no calibration); data risks (leakage, class imbalance, collection artifacts); lack of uncertainty handling (OOD detection); and the specifics of model training itself, which encourages spurious correlations and simplified decision rules (shortcuts). In response, we propose the discipline "audit → measure → improve", which combines explainability methods (*Explainable Artificial Intelligence*, XAI), representation-quality metrics, and out-of-distribution tests into a coherent, repeatable cycle: first we identify the "evidence" on which the model bases its decisions; next we quantify and verify it; finally, we introduce small, targeted modifications (based on previously detected problems) that minimize the cost in clean-data accuracy while improving the tool's practical utility.

Beyond the loop "audit → measure → improve", we propose a *portfolio* of methods for auditing and measuring the quality and robustness of representations: clustering as an early indicator of spurious correlations and the quality of semantic organization; OOD generalization and OOD detection; local and global XAI audits that reveal features potentially irrelevant causally; and multidimensional scaling together with a *numerical* assessment of projection quality. We further introduce adversarial evaluation based on

explainability and two improvement pathways for a model: (i) for text – contrastive finetuning and distillation supported by a *human-in-the-loop* protocol; (ii) for images – a lightweight loss function aimed at increasing the *Class Robustness Score* (understood as the fraction of attribution mass assigned to the annotated object) without degrading accuracy, while increasing robustness to black-box attacks. The whole is completed with practical guidance for open-world deployment of the produced models, enabling systematic assessment across the model lifecycle.

Having outlined the problem and introduced the methodology, we proceed to the first case study that illustrates the loop "audit → measure → improve" in practice. In Chapter 3, we propose a practical tool for vectorizing and classifying HTTP/URL traffic based on a language model. We then conduct a systematic XAI audit that exposes weaknesses in the solution. In parallel, we measure representation quality in clustering and OOD separation tasks, revealing issues invisible to standard classification metrics. Guided by the audit, we reshape the embedding geometry to improve representation quality, then rerun the measurements and compare them with the initial approach.

In the subsequent part of the dissertation, we propose *weaponizing* explainability, that is, deliberately repurposing explanation methods as tools for constructing adversarial attacks on text classifiers. The central idea is to use feature attributions as a mechanism for constraining the adversarial search space: instead of exploring the full combinatorial space of possible text perturbations, the attack operates only on those features that have the strongest influence on the model's decision. The proposed methods precisely modify – while preserving semantic meaning and grammatical correctness – those input components that are most important from the classifier's perspective, in order to induce misclassification. We consider both untargeted attacks (causing a transition to any other class) and targeted attacks (forcing prediction into a specific, chosen class). We demonstrate how this procedure applies both to classical text classifiers and to remotely accessible generative models, such as OpenChat, ChatGPT, and LLaMA, configured for *zero-shot* classification. In addition, we assess the perceptual quality of the generated adversarial samples in a human evaluation study.

We then propose a targeted method for improving the quality of visual representations in image classification models. Starting from a high-accuracy pretrained classifier, we audit its decisions in terms of the semantic validity of the evidence it relies on, assessed qualitatively through the correspondence between attribution maps and the ground-truth object. We adopt a localization-consistency measure – defined as the fraction of

attribution mass overlapping the annotated object – not only as a diagnostic tool, but as a training signal that directly guides model refinement. Building on this insight, we propose a lightweight, localization-aware training objective that reduces reliance on spurious background correlations and shifts representations toward object-centric features. The resulting procedure is architecture-agnostic and introduces only minimal additional supervision. It yields consistent improvements in localization consistency and, as a secondary effect, increases robustness to black-box adversarial attacks, while preserving classification performance on clean data.

This dissertation shows that the trustworthiness of models cannot be assessed by a single number – such as an F1 point or test accuracy. That still-dominant paradigm in the literature is too narrow to guarantee reliability in practice. Instead, models should be embedded in the cycle "audit → measure → improve", which makes system behavior visible in a broader perspective and enables targeted improvements. High values of classical metrics remain a good starting point, but they say little about whether – and which – model will perform best in a real application. Models designed with unexpected conditions in mind (including out-of-distribution settings and related tasks) fail less often and can be granted greater trust where the stakes are highest.

Streszczenie

Punktem wyjścia niniejszej rozprawy jest rozbieżność między „jednoliczbową” ewaluacją a realną wiarygodnością modeli, czyli ich zachowaniem w rzeczywistych warunkach. Wysoka trafność na danych należących do tego samego rozkładu co dane uczące (ang. *Independent and Identically Distributed*, IID) nie mówi, czy model opiera decyzję na właściwych przesłankach ani czy jego reprezentacje są użyteczne poza wąskim zadaniem. Nie wiemy też, jak model zachowa się w przypadku przesunięcia rozkładu lub pod wpływem drobnych, semantycznie nieistotnych zmian. W praktyce systemy trafiają w środowiska pełne korelacji pozornych, nowości i bodźców, które potrafią łatwo „oszukać” klasyfikator. Na tę lukę między testem a wdrożeniem składają się w szczególności: niestacjonarność i zmienność domeny; wąskie, niereprezentatywne procedury walidacyjne (brak oceny poza rozkładem uczącym i kalibracji); ryzyka związane z danymi (wycieki, niezrównoważenie klas, artefakty pozyskiwania); brak mechanizmów postępowania z niepewnością (detekcja OOD); a także specyfika uczenia modeli, która sprzyja korelacjom pozornym i uproszczonym regułom decyzyjnym. W odpowiedzi proponujemy dyscyplinę „audyt → pomiar → poprawa”, która łączy metody wyjaśnialności (ang. *Explainable Artificial Inteligence*, XAI), metryki jakości reprezentacji oraz testy poza-dystrybucyjne (ang. *Out-of-Distribution*, OOD) w spójny, powtarzalny cykl: najpierw rozpoznajemy, na jakich „dowodach” model opiera decyzje; następnie ilościowo to weryfikujemy; na końcu wprowadzamy małe, ukierunkowane modyfikacje, minimalizujące koszt w dokładności na danych czystych, a jednocześnie podnoszące praktyczną jakość narzędzia.

Poza samą pętlą „audyt → pomiar → poprawa” wprowadzamy *portfolio* metod do audytu i pomiaru jakości oraz odporności reprezentacji: klasteryzację jako wczesny wskaźnik korelacji pozornych i jakości organizacji danych; generalizację OOD oraz detekcję OOD; audyty XAI (lokalne i globalne) ujawniające cechy potencjalnie nieistotne przyczynowo; oraz skalowanie wielowymiarowe wraz z *liczbową* oceną jakości

rzutowania. Uzupełniamy to o ewaluację adwersaryjną z wykorzystaniem wyjaśnialności oraz o dwie ścieżki poprawy modeli: (i) dla tekstu – strojenie kontrastowe i destylacja wspierane protokołem *human-in-the-loop*; (ii) dla obrazu – lekka funkcja straty ukierunkowana na podniesienie miary zgodności lokalizacyjnej – rozumianej jako odsetek masy atrybucji przypadającej na adnotowany obiekt – bez pogorszenia trafności, a przy tym zwiększająca odporność na ataki czarnoskrzynkowe. Całość domyka praktyczny i zwięzły przewodnik, który syntezuje przedstawione metody w spójny, operacyjny schemat oceny i ciągłego doskonalenia modeli uczenia maszynowego w całym cyklu ich życia.

Po zarysowaniu problematyki i wprowadzeniu czytelnika w metodologię, przechodzimy do pierwszego studium przypadku, które ilustruje działanie pętli „audyt → pomiar → poprawa" w praktyce. W rozdziele trzecim, zaproponowano praktyczne narzędzie do wektoryzacji i klasyfikacji ruchu HTTP/URL oparte na modelu językowym RoBERTa. Następnie przeprowadzono systematyczny audyt XAI, który ujawnia słabe strony rozwiązania. Równolegle zmierzono jakość reprezentacji w zadaniach klasteryzacji oraz separacji OOD, co odsłoniło problemy niewidoczne z perspektywy standardowych miar jakości klasyfikacji. W dalszym kroku, kierując się wynikami audytu, przekształcono geometrię osadzeń, aby poprawić jakość reprezentacji, po czym ponownie przeprowadzono pomiary i porównano je z podejściem wyjściowym.

W dalszej części dysertacji proponujemy „zmilitaryzowanie" metod wyjaśnialności, czyli ich celowe wykorzystanie jako narzędzi do konstruowania ataków adwersaryjnych na klasyfikatory tekstowe. Kluczową ideą jest użycie atrybucji jako mechanizmu zawężania przestrzeni poszukiwań przykładów adwersaryjnych: zamiast eksplorować całą kombinatoryczną przestrzeń możliwych modyfikacji tekstu, atak koncentruje się wyłącznie na cechach o największym wpływnie na decyzje modelu. Zaproponowane metody modyfikują – przy zachowaniu poprawności znaczeniowej i gramatycznej – te cechy próbek, które są najbardziej istotne z perspektywy klasyfikacji, w celu wymuszenia błędu. Ataki realizujemy zarówno w trybie nieukierunkowanym (przestawienie do dowolnej innej klasy), jak i ukierunkowanym (do konkretnej, zadanej klasy). Pokazujemy, jak stosować tę procedurę zarówno wobec klasycznych klasyfikatorów tekstowych, jak i wobec modeli generatywnych dostępnych zdalnie, takich jak OpenChat, ChatGPT czy LLaMA, skonfigurowanych do klasyfikacji *zero-shot*. Dodatkowo oceniamy percepcyjną jakość wygenerowanych próbek w badaniu z udziałem ludzi.

Następnie przechodzimy do propozycji ukierunkowanej metody poprawy jakości

reprezentacji wizualnych w modelach do klasyfikacji obrazów. Punktem wyjścia jest model o wysokiej trafności, którego decyzje audytujemy pod kątem sensowności używanych „dowodów", rozumianej jako zgodność atrybucji z rzeczywistym obiektem. Wykorzystujemy miarę zgodności lokalizacyjnej – definiowaną jako odsetek masy atrybucji pokrywającej się z adnotowanym obiektem – nie tylko jako narzędzie diagnostyczne, lecz jako sygnał sterujący procesem uczenia. Na tej podstawie proponujemy nową funkcję straty, która ogranicza poleganie na korelacjach kontekstowych i przesuwa reprezentacje w stronę cech obiektowych. Zaproponowana procedura stanowi lekki i architektonicznie agnostyczny mechanizm poprawy reprezentacji. Prowadzi ona do istotnej poprawy modelu z perspektywy miary zgodności lokalizacyjnej i odporności na przeprowadzane ataki adwersaryjne bez wiedzy o modelu (ang. *black-box adversarial attacks*), przy zachowaniu jakości na danych czystych.

Rozprawa pokazuje, że wiarygodności modeli nie da się ocenić wyłącznie jedną liczbą – na przykład punktem F1 ani samą trafnością testową. Ten, wciąż dominujący w literaturze, paradygmat jest zbyt wąski, by gwarantować niezawodność w praktyce. Zamiast tego modele należy osadzać w cyklu „audyt → pomiar → poprawa", który pozwala zobaczyć działanie systemu w szerszej perspektywie i wprowadzać celowane ulepszenia. Wysokie wartości klasycznych metryk pozostają oczywiście dobrym punktem startowym, lecz niewiele mówią o tym, czy i który model najlepiej sprawdzi się w realnej aplikacji. Modele projektowane z myślą o zachowaniu w warunkach niespodziewanych (w tym poza dystrybucją oraz w pokrewnych zadaniach) rzadziej zawodzą i mogą być obdarzane większym zaufaniem tam, gdzie stawka jest najwyższa.

Contents

Chapter 1

Introduction

1.1 Motivation

Building a reliable machine learning system requires more than high test accuracy. Models that achieve strong performance on standard aggregate evaluation metrics may still rely on spurious correlations, fail under distribution shifts, or justify decisions with evidence that is misaligned with the task goal. We argue that trustworthiness (understood as the joint pursuit of performance, transparency, and robustness) demands that explainability and representation quality be regarded as integral components of any Machine Learning pipeline.

In practice, explanations help us to audit what evidence a model actually relies on when making predictions. Yet, we need quantitative, qualitative, and repeatable checks that (i) reveal brittle or shortcut features, (ii) diagnose whether the learned embedding space meaningfully organizes data (e.g. supports semantically coherent clustering of samples and separates out-of-distribution samples), and (iii) link these properties to practical ways of improving models with small, targeted interventions that modify representations through lightweight fine-tuning rather than full retraining. This motivates a unified workflow: *audit* baselines with XAI → *measure* representation quality → *improve* with active learning / fine-tuning. We apply the same logic across modalities (text and images) while adapting the audit signals and evaluation criteria to modality-specific reliability cues.

Recent cross-domain benchmarks reinforce this position: when models are probed under multiple orthogonal stressors, including common corruptions, adversarial per-

turbations, novelty (Out-of-Distribution, OOD), and even unrecognizable "nonsense" inputs, the apparent robustness of state-of-the-art systems fragments into a mosaic of strengths and blind spots. In vision, comprehensive testbeds show that improving one robustness axis can degrade another, and that high clean accuracy frequently coexists with brittle (unpredictable or degraded) behavior [1]–[4]. In NLP, similar patterns emerge under lexical shifts, paraphrases, and character-level noise [5], [6]. The lesson is that reliability must be evaluated as a portfolio of behaviors, not as a single scalar.

In high-stakes settings, the cost of brittle or unexpected model behavior is not a single point on a leaderboard but real harm. In medicine, diagnostic support systems must highlight task-relevant evidence rather than spurious markers, generalize across scanners and hospitals, and explicitly handle unfamiliar or out-of-distribution cases; otherwise, they risk missed diagnoses or over-treatment [7]–[9]. In autonomous driving, perception stacks need to remain reliable under weather, sensor drift, and rare hazards, with explanations that localize the cues behind decisions such as "stop" or "yield", so failures can be audited before they recur [10], [11]. In everyday knowledge work, foundation-model assistants increasingly mediate search, drafting, and analysis for millions of users; when their representations encode shortcuts, entire downstream workflows inherit those biases [12]. These pressures make it essential to evaluate not only accuracy, but also how models arrive at their decisions. This requires explanations that reveal evidence use, as well as embedding spaces whose geometry supports rejecting out-of-distribution inputs, similarity-based retrieval, unsupervised structure discovery through clustering and monitoring representation drift over time.

Historically, the limitations of accuracy as a main indicator of model reliability crystallized through two lines of work. First, adversarial examples revealed high-confidence errors in dense neighborhoods of the training data manifold [13], [14], underscoring a discrepancy between human and model invariances. Second, work on out-of-distribution (OOD) detection reframed reliability as the ability to confidently accept in-distribution inputs while rejecting inputs that fall outside the training distribution [15]–[17]. Together, these threads shifted the focus from accuracy under closed-set assumptions toward the structure and behavior of representations under stress, laying the groundwork for representation-centric evaluation.

A similar shift is underway in governance and risk management. Guidance such as the NIST AI Risk Management Framework and the EU AI Act calls for documented evaluation under distribution shift, clear statements of model limits, and continuous

monitoring. In practice, this turns the audit loop (audit → measure → improve) from a "nice to have" into a requirement [18], [19]. Leading researchers have also urged greater focus on safety and evaluation, warning that capability gains are outpacing the rigor of tests (e.g. [20]–[22]). This dissertation answers this call at the technical level: it defines trustworthiness as concrete, measurable properties of explanations and representations, and it shows how to act on those measurements.

Concretely, we treat explanations as diagnostics of evidence use – understood as the features, tokens, or regions that contribute causally to a model's prediction – and representations as the shared substrate on which multiple reliability properties depend. We argue that trustworthiness cannot be assessed through a single performance metric but requires a portfolio of complementary signals that probe how models use evidence, how representations are structured, and how predictions degrade under distribution shift, perturbations, and adversarial stress. This perspective motivates a unified *audit-measure-improve* loop, in which models are systematically inspected, evaluated, and incrementally refined. We do not claim to solve trustworthiness end-to-end; instead, we advocate a practical, iterative approach in which identified failure modes are not only made visible, but are directly linked to targeted interventions that demonstrably improve representation quality and robustness.

Problem formulation. This dissertation is motivated by the observation that machine learning systems are still predominantly judged by aggregate classification metrics, which can mask important deficiencies in how models use evidence and how their internal representations behave outside of training conditions. In response, we investigate representation-centric evaluation measures that expose such gaps and study lightweight, targeted interventions that reshape embedding geometry without sacrificing predictive performance.

We start from inputs drawn from different data modalities (e.g. text or images) and learn two coupled objects: compact internal representations and predictors built on top of them. The representations encode the input in a latent space, while the predictor maps this representation to a class decision by selecting the highest-scoring label.

Trustworthiness requires such models to satisfy three desiderata simultaneously: **(A) Accuracy** on standard held–out test data drawn IID from the training distribution, **(B) Auditability** through faithful and interpretable explanations, and **(C) Robustness** to distribution shifts and to input perturbations.

To make desiderata (B) and (C) concrete, we operationalize them using observable

signals derived from explanations and representations. Explanations are used primarily as diagnostics of *evidence use*, that is, to assess whether model decisions rely on task-relevant features or instead exploit spurious correlations present in the data. Representations are examined as geometric objects: we analyze whether the embedding space exhibits meaningful structure, such as semantic organization, neighborhood consistency, and separation between in-distribution and out-of-distribution samples.

While our primary focus is on representations learned end-to-end by modern models, the proposed audits and measurements apply equally to fixed, expert-designed feature spaces. In such cases, the representation is not learned but provided a priori; nevertheless, clustering structure, OOD separability, explanation alignment, and robustness under perturbations remain well-defined and diagnostically useful. Crucially, the insights obtained from these audits can guide domain experts in refining such hand-crafted features, for example by identifying redundant, spurious, or overly context-dependent attributes and motivating their removal, reweighting, or redesign. In this sense, the framework supports both automated representation learning and expert-driven feature engineering.

Robustness is further assessed by studying model behavior under distribution shifts and under targeted, semantics-preserving perturbations. Taken together, these signals characterize not only predictive performance, but also how models arrive at their decisions and how stable those decisions remain under stress. This perspective places the quality of the representation at the center of the analysis and motivates the study of methods that explicitly reshape the embedding space to improve reliability beyond standard accuracy metrics.

Hypothesis. The central hypothesis of this dissertation is that the trustworthiness, robustness, and safety of machine learning models deployed in real-world settings – where previously unseen or out-of-distribution situations may occur – can be improved through a systematic, multi-criteria audit of model quality that enables incremental refinement of the learned representations.

Before stating the concrete contributions that substantiate this hypothesis, we introduce a set of working definitions to disambiguate terminology that is used inconsistently across the literature.

Working definitions. The vocabulary around reliability is used inconsistently in the literature, so we fix it here to keep later chapters precise (based on [1]). By *IID*

generalization we mean the familiar setting where training and test samples are drawn from the same distribution; accuracy on a fresh held-out split captures this. In the wild, data rarely match training perfectly; this is *distribution shift*. We will call samples similar to training *in-distribution* (ID) and everything else *out-of-distribution* (OOD), acknowledging that shifts may affect inputs (covariate shift), labels (label shift) or the relation between them (concept shift). *OOD generalization* asks whether a model remains useful when the test data are plausibly related to the task but distributed differently (new sources, time periods, styles); we informally distinguish *near-OOD* (subtle changes) from *far-OOD* (obvious domain gaps). Separate from that is *OOD detection*: given an ID reference set, decide whether a new sample looks ID or OOD without knowing its correct class. We summarize this with AUROC/AUPR and operating points such as TNR@95%. Put simply: *generalization* asks whether the model remains accurate on the same task, while *OOD* asks whether an input even belongs to the training distribution, so a model can classify *near-OOD* correctly if classes overlap, whereas OOD detection must also flag samples with novel classes or unrelated content. We use *robustness* to mean that predictions do not flip under small, semantics-preserving changes to the input (*natural robustness*); when changes are chosen by an attacker within a bounded budget we speak of *adversarial robustness*. Because explanations are first-class citizens in this thesis, we also care about *explanation robustness*. By this we mean that attributions (i) remain stable under small, label-preserving edits and (ii) concentrate on task-relevant evidence (e.g. true attack patterns in text or annotated objects in images). However, XAI is used here primarily as a diagnostic and as an adversarial-guidance tool rather than as a sole measure of robustness. Finally, *auditability* denotes the ability to inspect and contest decisions via short, repeatable checks that pair attribution views with quantitative signals from the representation (cluster structure, neighborhood consistency, and OOD separability).

1.2 Main contributions

Guided by the central hypothesis of this dissertation – that the trustworthiness and robustness of machine learning models can be improved – this dissertation makes the following concrete contributions, corresponding to the research objectives achieved in the course of the conducted studies. The contributions span methodology, evaluation, and practical workflows for trustworthy machine learning across text and vision. A

substantial part of this work has been previously presented in international peer-reviewed publications. A complete list of publications is provided in the Appendix A.

1. **A systematic audit–measure–improve procedure for model reliability.** We formalize a systematic and iterative procedure for assessing and improving the trustworthiness of machine learning models, structured as an *audit* → *measure* → *improve* loop. The procedure integrates qualitative and quantitative analysis of evidence use, representation quality, and boundary behavior, and enables incremental, targeted refinement of learned representations rather than full model retraining. Through extensive empirical evaluation on both text and image data, we show that this procedure can substantially improve model interpretability and robustness, including adversarial robustness, thereby increasing the practical value of deployed ML systems.

2. **A portfolio of methods for auditing and measuring representation quality and robustness.** As part of the proposed audit–measure–improve procedure, we introduce a portfolio of complementary methods for evaluating representation quality and model robustness. These include clustering-based diagnostics for detecting shortcut learning and spurious correlations, measures of out-of-distribution (OOD) generalization and detection, explainability-based checks of evidence alignment, and geometry-based analyses of embedding spaces. Together, these methods provide a multi-aspect, representation-centric assessment that goes beyond single-number classification metrics.

3. **Adversarial evaluation via explanation-guided perturbations.** We extend the evaluation portfolio with adversarial testing and introduce an explanation-guided attack methodology, referred to as *weaponized XAI*. In this setting, attribution methods are used to identify high-impact features that guide targeted, semantics-preserving perturbations. Robustness to such attacks provides complementary evidence of model reliability beyond standard accuracy and representation diagnostics. We validate the approach on conventional text classifiers as well as on generative models configured for zero-shot classification, including systems accessed via remote APIs.

4. **Targeted methods for improving representation quality in text and vision.** Guided by the findings of the audit and measurement stages, we propose targeted

methods for improving representation quality in both text and image domains. For text-based models, we reshape embedding geometry using contrastive finetuning and representation distillation supported by a human-in-the-loop protocol, and quantify the amount of human feedback required to achieve measurable improvements. For vision models trained on large-scale datasets such as ImageNet, we introduce a localization-aware training objective that reduces reliance on spurious background correlations while preserving classification accuracy. Crucially, these interventions are applied in a targeted manner, focusing on systematically underperforming classes rather than uniformly retraining across all labels – a particularly challenging setting for models operating over hundreds of classes.

5. **Empirical case studies demonstrating practical applicability.** We conduct a series of empirical case studies that illustrate the practical effectiveness of the proposed audit–measure–improve methodology. In particular, we demonstrate that the Sec2Vec approach, developed within the framework introduced in this dissertation, significantly improves the robustness and interpretability of machine learning models used for detecting malicious network traffic based on textual protocol data.

6. **Practice-oriented guidance and a unified lifecycle checklist.** Throughout the dissertation, we distill practical insights obtained from the conducted studies into actionable guidance for deployment and monitoring. These insights are consolidated into a concise lifecycle checklist that helps keep deployed systems aligned with the *audit* → *measure* → *improve* loop and supports continuous evaluation under distribution shift.

Thesis layout and roadmap

This dissertation is organized to progressively develop, evaluate and apply the proposed audit–measure–improve methodology across multiple modalities.

Foundations. Chapter 2 reviews the conceptual and technical background related to representation learning, explainability (XAI), out-of-distribution detection, and robustness. It also introduces the notation, definitions, and evaluation principles reused throughout the dissertation. This chapter establishes the conceptual background and evaluation principles that support Contributions (1) and (2).

Improving the quality of the representation in text. Chapter 3 presents a text-based case study focused on HTTP/URL classification. It introduces the baseline pipeline, performs a systematic audit and evaluation of representations, and proposes targeted methods for improving representation quality and robustness. This chapter addresses Contributions (1), (2), and (5) by instantiating the audit–measure–improve procedure for text representations and proposing methods for improving their quality and robustness.

Explanation-guided adversarial evaluation. Chapter 4 extends the analysis to adversarial settings, using proposed explanation-guided adversarial attacks to stress-test models and assess robustness beyond standard evaluation scenarios. This chapter primarily supports Contribution (3) and (4) by extending the evaluation portfolio.

Improving representation quality in vision. Chapter 5 applies the introduced methodology to image classification and proposes a method to improve the quality of representation in deep vision models. The obtained results are also evaluated under adversarial stressors. This chapter realizes Contribution (5) in the vision domain and provides additional empirical evidence for Contributions (1) and (2) through localization-aware robustness audits and improvements.

Summary and formulation of recommendations. Chapter 6 synthesizes insights from all the chapters, summarizes and discusses the key results, and formulates practical recommendations for the development and monitoring of trustworthy machine learning systems.This chapter realizes Contribution (6) and synthesizes Contributions (1)-(5).

Chapter 2

Theoretical Background

Before presenting the empirical results, we introduce the theoretical and methodological background that the rest of the dissertation relies on. The goal is not to be exhaustive, but to establish a precise vocabulary and a compact toolkit of models, objectives, and evaluation measures that will be instantiated and audited in later chapters.

2.1 From Classical Machine Learning to Modern Deep Models

Let us begin with one of the most fundamental branches of machine learning, *supervised learning*. We assume access to a labeled dataset

$$\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{n}, \quad x_i \in \mathcal{X}, \quad y_i \in \{1, \dots, C\},$$

where each input x_i belongs to the input space $\mathcal{X}$ and is associated with a ground-truth label y_i drawn from C possible classes.

In line with the representation-centric perspective adopted throughout this dissertation, we decompose prediction into two stages. First, an input x is mapped to a representation

$$z = f_\theta(x),$$

where f_θ may correspond either to a learned encoder or to a domain-specific, expert-defined feature extractor. Second, a classifier g_ϕ maps the representation to a vector of

class-wise confidence scores

$$g_\phi(z) \in \mathbb{R}^C.$$

These scores can, if needed, be converted into class probabilities ($P(y \mid z)$) via a suitable normalization (e.g. softmax), but for decision making only their relative ordering matters. The final prediction is obtained by selecting the class with the highest score:

$$y = \arg\, max_{c\in\{1,\dots,C\}}\ g_\phi(f_\theta(x))_c.$$

Training is typically performed by minimizing the *empirical risk* over the composed model $h_{\theta,\phi} = g_\phi \circ f_\theta$:

$$\min_{\theta,\phi}\ \frac{1}{n}\sum_{i=1}^{n} \ell\big(g_\phi(f_\theta(x_i)),\ y_i\big)\ +\ \Omega(\theta,\phi), \tag{2.1}$$

where ℓ is a loss function (commonly the cross-entropy loss) and Ω denotes a regularization term that penalizes overly complex models. For multi-class classification, the cross-entropy loss takes the form

$$\ell\big(g_\phi(f_\theta(x_i)), y_i\big) = -\log g_\phi(f_\theta(x_i))_{y_i}.$$

Regularization can be interpreted as a form of "simplicity bias": it discourages the model from fitting noise in the training data by penalizing large parameter values (e.g. ℓ_2 regularization) or encouraging sparsity (e.g. ℓ_1 regularization). Without such mechanisms, highly flexible models could memorize the training data instead of learning generalizable patterns. In practice, the objective in (2.1) is optimized in an iterative training loop: at each step, the model produces predictions, the loss is computed, and gradients are used to update the parameters in the direction of lower loss. Over many such iterations, the predictor gradually improves its ability to generalize to unseen data.

Evaluation. For evaluation, the dataset is commonly split into three parts: (i) the *training set*, used to optimize the parameters, (ii) the *validation set*, used for model selection and hyperparameter tuning, and (iii) the *test set*, reserved for the final performance assessment. To ensure that results are not due to a specific split, one often applies *cross-validation*, which repeatedly partitions the dataset into training and validation folds and averages the outcomes. Since most learning algorithms rely on randomized

initialization, it is also common to repeat training several times (within computational limits) to account for stochastic variability.

The effectiveness of supervised classifiers is typically measured by *accuracy*, i.e. the proportion of correctly classified examples. However, this metric can be misleading in the presence of class imbalance. Consider, for example, a dataset where 95% of samples belong to a single class: a trivial classifier that always predicts this majority class will reach 95% accuracy, yet provide no real utility.

A more informative picture is obtained from the *confusion matrix*, which records how many samples of each true class were assigned to each predicted class. In the binary case, this yields four quantities: *true positives* (TP), *true negatives* (TN), *false positives* (FP), and *false negatives* (FN). From these counts, one can define common evaluation scores:

- **Precision** = TP / (TP + FP), measuring the fraction of predicted positives that are truly positive.

- **Recall** = TP / (TP + FN), measuring the fraction of actual positives that were correctly detected.

- **F1-score**, the harmonic mean of precision and recall, balancing both aspects in a single metric.

When classifiers produce probabilistic scores rather than hard decisions, thresholding these scores yields different trade-offs between FPR and recall (TPR). The *receiver operating characteristic* (ROC) curve (Figure 2.1 summarizes this trade-off by plotting the true positive rate (recall) against the false positive rate across thresholds. For example, when a classifier outputs class probabilities, adjusting the decision threshold biases the model toward one class or another, thereby changing the balance between missed detections and false alarms. The *area under the ROC curve* (AUROC) condenses this information into a single number between 0 and 1. Higher AUROC means that the classifier ranks positive examples consistently above negative ones. In addition to the area, it is often useful to report *FPR@95%*, the false positive rate at a fixed true positive rate (95% recall). This quantity highlights how many false alarms must be accepted in order to detect nearly all true positives.

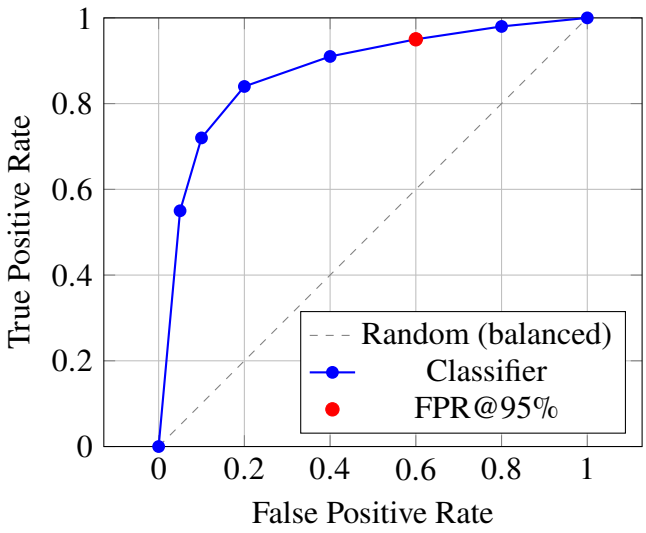


Figure 2.1: Example ROC curve. The dashed diagonal corresponds to random guessing under a balanced dataset. The blue curve shows a classifier with AUROC ≈ 0.92. The red point highlights the false positive rate at 95% recall (FPR@95%).

2.1.1 Classical Representations and Predictors

The most fundamental way of representing data is in a *tabular form*. Each sample is encoded as a fixed length *feature vector*, whose components can correspond to named attributes (e.g. length, width, counts, binary flags) or remain anonymous numerical features. In the supervised learning setting, every such sample is paired with a label y_i (a class). Conceptually, this representation treats data points directly as coordinates in a vector space. Once expressed in this way, a wide range of learning algorithms can be applied directly on the feature vectors; some of them are discussed below.

Naive Bayes (NB)

One of the simplest, yet theoretically elegant classifiers is *Naive Bayes* (NB) [23]. It is based on Bayes' theorem, which relates the conditional probability of a class y given features x:

$$P(y \mid x) = \frac{P(x \mid y)\,P(y)}{P(x)}.$$

Here, $P(y)$ is the prior probability of a class, $P(x \mid y)$ is the likelihood of observing features x under class y, and $P(x)$ is a normalization constant. The classifier assigns a

new sample to the most probable class:

$$h(x) = \arg\max_{y \in \{1,\dots,C\}} P(y) \prod_{j=1}^{d} P(x_j \mid y).$$

The key assumption (hence "naive") is that features $\{x_1, \dots, x_d\}$ are conditionally independent given the class y. Although rarely true in real-world data, this assumption greatly simplifies estimation: one only needs to learn one-dimensional distributions for each feature within each class. If the assumption were exactly correct and the feature distributions were known, NB would be *optimal*, achieving the lowest possible error (the Bayes error). In practice, despite the strong independence assumption, NB often works surprisingly well as a baseline, especially for sparse and high-dimensional data. Figure 2.2 illustrates the Naive Bayes decision rule in a one-dimensional setting, where class-conditional densities can be visualized directly.

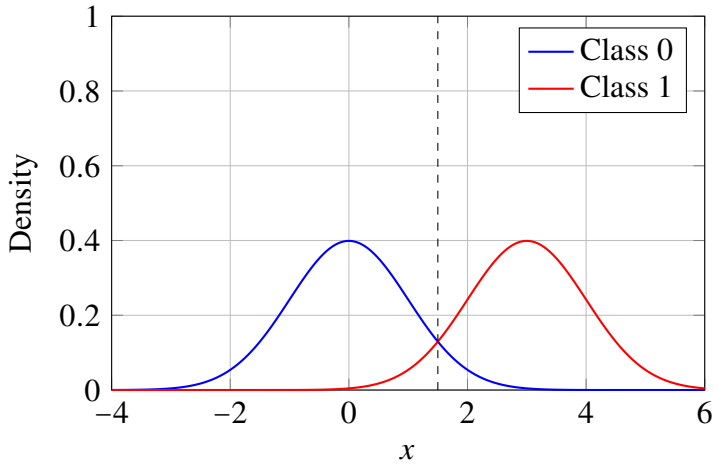


Figure 2.2: Illustration of Naive Bayes classification with Gaussian likelihoods for two classes. The dashed vertical line marks the decision boundary where posterior probabilities are equal.

Logistic Regression (LR)

Another widely used baseline classifier is *logistic regression* (LR) [24]. Despite its name, it is a classification method. The idea is to start from a linear model:

$$z(x) = w^\top x + b,$$

which maps the feature vector $x \in \mathbb{R}^d$ onto a real value. Instead of using $z(x)$ directly for prediction, we pass it through the *logistic sigmoid function*:

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

The output $\sigma(z(x))$ can be interpreted as the probability of the positive class. A decision is made by thresholding this probability, typically at 0.5. Thus, logistic regression learns a linear *decision boundary*, but instead of hard rules, it models a smooth transition between classes (see Figure 2.3).

In practice, the parameters (w, b) are found by minimizing the cross-entropy loss. This makes logistic regression both simple and interpretable: coefficients w_j indicate the importance of each feature in favoring one class over the other. Despite its simplicity, LR remains competitive for many tabular datasets and serves as a powerful baseline.

Although both Naive Bayes (NB) and Logistic Regression (LR) can be applied to the same classification tasks, they rely on fundamentally different modeling principles. NB is a *generative* model: it explicitly models the data-generating process by specifying the class prior $P(y)$ and the class-conditional distribution $P(x \mid y)$, typically under the simplifying assumption that features are conditionally independent given the class. Predictions are then obtained by applying Bayes' theorem to compute $P(y \mid x)$. In contrast, LR is a *discriminative* model: it does not model the data distribution, but directly parameterizes the conditional probability $P(y \mid x)$.

If the modeling assumptions of Naive Bayes – namely conditional independence and correctly specified class-conditional distributions – were exactly satisfied, NB would be theoretically optimal, achieving the minimum possible classification error (the Bayes error). In practice, these assumptions are rarely met. As a consequence, NB tends to perform well in high-dimensional and sparse settings, where the independence assumption is a reasonable approximation, whereas Logistic Regression typically performs better when features are correlated or when predictive accuracy is the primary objective.

Decision Trees.

A *decision tree* [25] is a classifier that recursively partitions the feature space into regions aligned with the axis. At each internal node, a threshold is used on a single feature to split the data, and the leaves correspond to the predictions of the final class. Trees

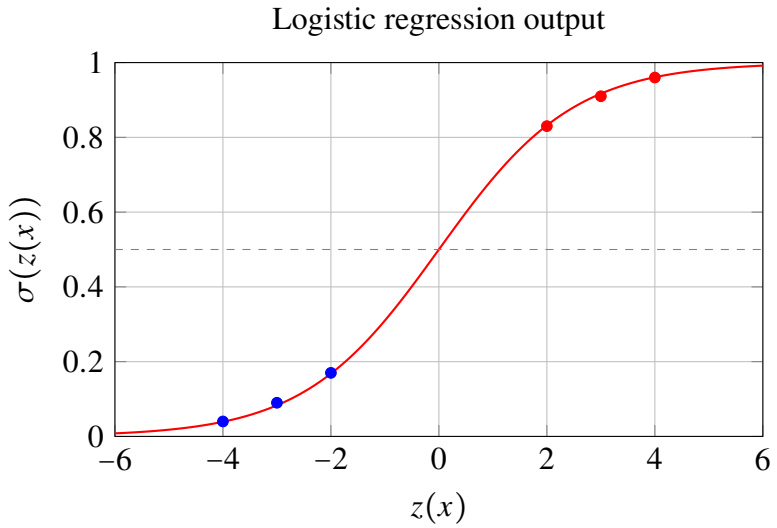


Figure 2.3: Logistic regression, mapping unbounded values from linear model into probabilities via the sigmoid. Blue/red points illustrate samples from two classes.

are learned greedily by choosing splits that maximize class purity (e.g. information gain or Gini index). The resulting model is highly interpretable: each decision path corresponds to a "if-then" rule. A *random forest* [26] combines many decision trees into an ensemble. Each tree is trained on a bootstrap sample of the data and considers a random subset of features at each split. The forest aggregates their predictions, usually by majority vote. This procedure reduces variance and mitigates overfitting, making random forests more robust than single trees. The visualization of both approaches is shown in Figure 2.4.

Unlike Naive Bayes, which assumes feature independence, or Logistic Regression, which imposes a linear boundary, decision trees can carve out non-linear, axis-aligned regions of arbitrary complexity. Random forests further improve generalization by averaging across many diverse trees. As a result, they often achieve strong performance on tabular data, though at the cost of reduced interpretability (yet to be discussed later) compared to a single tree (see 2.4).

From Perceptron to Multilayer Perceptron (MLP)

The starting point for neural networks is the *perceptron*, introduced by Rosenblatt in 1958 [27]. A perceptron takes inputs $x_1, \ldots, x_d$, multiplies them by weights $w_1, \ldots, w_d$,

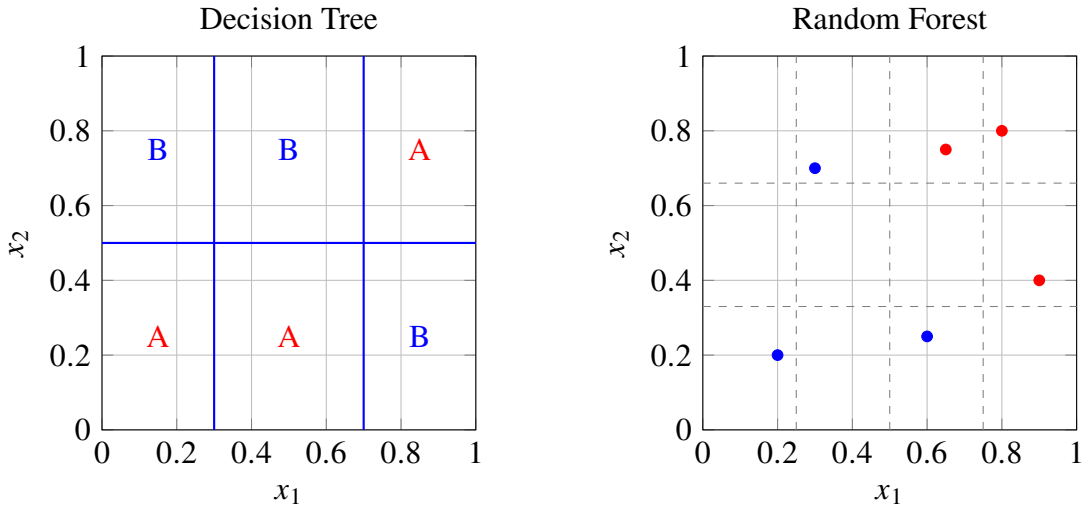


Figure 2.4: Decision tree (left): axis-aligned partitions with simple rules. Random forest (right): multiple diverse partitions (dashed) aggregated by voting, producing more robust class assignments.

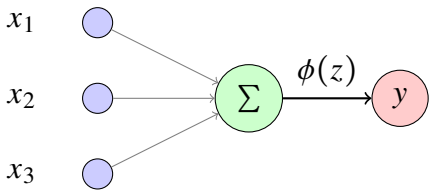


Figure 2.5: A single perceptron: weighted inputs are summed, bias added, and an activation function determines the output.

adds a bias b, and applies an activation function:

$$z = \sum_{j=1}^{d} w_j x_j + b, \quad y = \phi(z).$$

With a step function ϕ, it produces a binary decision based on a linear boundary. Perceptrons can separate linearly separable data but fail on problems like XOR, which require non-linear decision surfaces. The visualization of perceptron is given in Figure 2.5.

The natural extension is *multilayer perceptron* (MLP) [28], also called a fully connected feedforward network. By stacking perceptrons into hidden layers and using

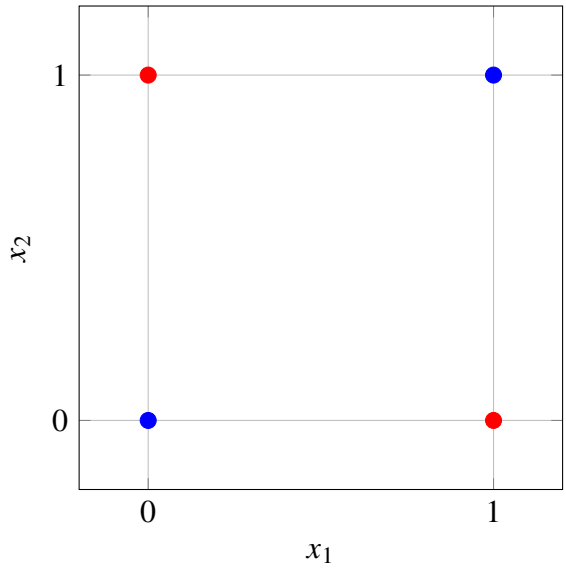


Figure 2.6: The XOR problem: blue and red points at the corners of a square cannot be separated by any single linear boundary. MLPs overcome this by combining hidden units to construct non-linear decision surfaces.

non-linear activations, MLPs can represent complex decision boundaries. The universal approximation theorem shows that an MLP with enough hidden units can approximate any continuous function, making it a flexible and general model. Training such networks is performed iteratively by minimizing a *loss function*, which quantifies the error between the predictions and the true labels. Common choices are cross-entropy for classification and mean squared error (MSE) for regression. Optimization proceeds through *gradient descent*, where the error in the output layer is propagated back through the network using the *backpropagation* algorithm, allowing the loss gradients with respect to all weights to be efficiently calculated and the parameters to be updated in stochastic steps.

XOR problem and comparison. A canonical example that highlights the differences between classifiers is *XOR problem*. Data points are placed at the corners of a square: class A at $(0,0)$ and $(1,1)$, and class B at $(0,1)$ and $(1,0)$. No single linear boundary can separate the two classes, so methods like Naive Bayes or Logistic Regression fail. Decision Trees or Random Forests can solve XOR by axis-aligned splits, but in a fragmented way. A multilayer perceptron, however, can combine hidden units to form the required non-linear boundary in a smooth manner.

2.2 Text representations: from language to vectors

We have seen how machine learning algorithms can be applied to tabular data, where each sample is represented as a fixed-length feature vector. Extending these methods to text requires an appropriate choice of representation. Unlike numerical data, text consists of sequences of tokens of varying length, which cannot be used directly by standard classifiers. To apply models such as logistic regression, random forests, or neural networks, textual inputs must first be mapped to numerical feature vectors that capture their content.

This section briefly reviews the evolution of text representations, from classical count-based models (Bag-of-Words, TFIDF) and static word embeddings to modern transformer-based encoders, in order to position the representations studied in this dissertation. While the classical methods are included as historical baselines, the focus of this thesis is on *contextual representations produced by pretrained transformer encoders*. In subsequent chapters, these encoder outputs serve as the primary objects of analysis: their interpretability, robustness, and overall quality are systematically audited, measured, and improved through targeted fine-tuning, distillation, and human-in-the-loop protocols.

2.2.1 Bag-of-Words and TFIDF

A classical approach is the *Bag-of-Words* (BoW) model [29]. Here, each document is represented by a vector whose components count how many times each word from the vocabulary (the most frequently occurring words in a dataset) appears. Order and syntax are discarded, but BoW provides a simple way to map text into the vector space required by machine learning algorithms. An example of a document represented using this method is shown in Figure 2.7.

A refinement is *Term Frequency Inverse Document Frequency* (TFIDF) [30], which rescales raw counts. Term frequency reflects how often a word appears in a document, while inverse document frequency downweights words that are too common across the corpus. The resulting weight is

$$\text{tfidf}(t,d) = \text{tf}(t,d) \cdot \log \frac{N}{df(t)},$$

where $\text{tf}(t,d)$ is the frequency of the term t in document d, N is the number of documents and $df(t)$ is the number of documents containing t. Compared to BoW,

TFIDF highlights distinctive words that help discriminate between documents.

These classical vectorization schemes are easy to implement, efficient, and still strong baselines for many text classification tasks. However, they ignore word order and semantics, motivating more advanced representation methods, which we will discuss next. Another limitation is the problem of *out-of-vocabulary* (OOV). If a word is not present in the training corpus, it will not appear in the vocabulary and, therefore, cannot be represented in the vector space. This leads to information loss, since all unseen words are effectively treated as unknown tokens.

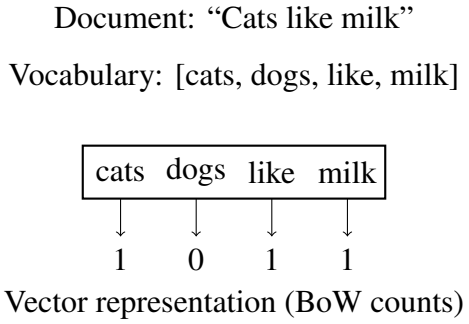


Figure 2.7: Example of Bag-of-Words representation: the document “cats like milk” is mapped into a fixed-length vector by counting word occurrences from the vocabulary.

2.2.2 Word2Vec and FastText

While Bag-of-Words and TFIDF treat documents as collections of words, they do not capture semantic relationships. A major breakthrough came with *Word2Vec* [31], which learns dense vector embeddings for individual words. Each word is represented as a point in a continuous vector space such that words with similar meaning appear close to each other. This enables capturing syntactic and semantic patterns that are inaccessible to simple count models.

Word2Vec is an example of *unsupervised representation learning*. The model does not require explicit labels but learns embeddings by predicting words from their context in raw text. Two architectures are commonly used (Figure 2.8):

- **Skip-gram:** predict surrounding context words given a central word w_t:

$$\max_{\theta} \frac{1}{T} \sum_{t=1}^{T} \sum_{-c \le j \le c,\, j \neq 0} \log P(w_{t+j} \mid w_t; \theta),$$

  where T is the text length, c is the context window size, and $P(\cdot \mid w_t; \theta)$ is the probability of context words given the central word.

- **CBOW:** predict the central word given its surrounding context:

$$\max_{\theta} \frac{1}{T} \sum_{t=1}^{T} \log P(w_t \mid w_{t-c}, \ldots, w_{t-1}, w_{t+1}, \ldots, w_{t+c}; \theta).$$

In both cases, the probability $P(\cdot)$ is parameterized by a simple neural network: a one-hot input is mapped through a hidden embedding layer to a low-dimensional vector, and the output layer uses softmax to predict target words. Initially, word vectors are random; during training they are refined by stochastic gradient descent, and the hidden layer embeddings become the final distributed representations.

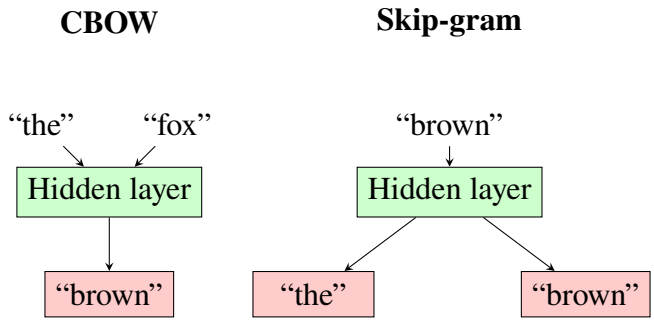


Figure 2.8: Word2Vec architectures: CBOW averages context embeddings to predict the central word, while Skip-gram uses the central word to predict its surrounding context. Both optimize embeddings in an unsupervised manner by maximizing the likelihood of observed word co-occurrences.

A key limitation of Word2Vec is the *out-of-vocabulary* (OOV) problem: words unseen during training cannot be embedded. Moreover, it treats words as atomic units, ignoring their internal structure. *FastText* [32] addresses these issues by representing each word as a composition of its character n-grams. For example, the word “playing” is represented as the sum of embeddings for <*pla*, *lay*, *ayi*, *yin*, *ing*>. This allows

embeddings to be generated even for unseen words, as long as their subword units were observed. By modeling morphological structure, FastText proves especially useful for highly inflected languages. Training follows the same Skip-gram or CBOW framework as Word2Vec, but the embedding of a word is computed from its subword vectors (average).

Both of the above methods share an interesting property. Since the meaning of each word is obtained from its surrounding context, the resulting embedding space often exhibits a remarkable structure: not only individual vectors, but also their differences carry semantic meaning. Figure 2.9 illustrates the classical example with “king” and “queen”, where vector offsets correspond to semantic relations. Another well-known case is that “Warsaw” should relate to “Poland” in the same way that “Paris” relates to “France”. Moreover, words frequently used in similar contexts tend to cluster together, such as terms related to sweets (“cookies”, “candies”, etc.). This demonstrates that the feature space learned by such models can capture meaningful semantic and syntactic relations.

This, however, does not yet provide us with document-level vectors that can be directly used for classification. A standard approach is to compute a simple average of all word embeddings that occur in a given text. Unfortunately, averaging tends to wash out important information and the resulting representation space is often not particularly discriminative. Here, transformer-based models come to the rescue, as they are designed to produce contextualized embeddings for entire sentences or documents.

2.2.3 Transformers

The *Transformer* [33] was introduced as an encoder-decoder model for machine translation. The encoder (left in Figure 2.10) maps an input sentence to contextual embeddings; the decoder (right) generates the target sentence, one token at a time, conditioned on the output of the encoder. For text representation and classification, we usually keep *only the encoder stack*, while generative language models use *only the decoder stack*. Unlike RNNs (for example, LSTMs) that read tokens sequentially, transformers process all tokens in parallel via *self-attention*, capturing long-range dependencies in a single step and enabling efficient training.

We now move from the high-level picture to a component-wise description of

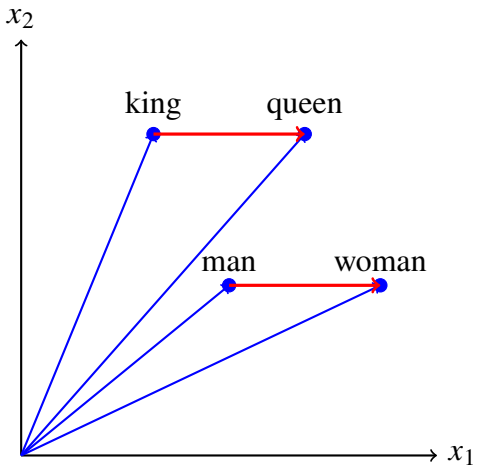


Figure 2.9: Classic word embedding analogy: the offset "king"→"queen" is parallel to "man"→"woman". Word embeddings capture semantic relations as linear translations in vector space.

the transformer encoder. At each step we explain what the block does, how it is parameterized, and how its parameters are learned during training. We start with *input embeddings*.

Input embeddings

The text is first tokenized as a sequence of discrete symbols. These are typically *subword* units obtained with Byte-Pair Encoding (BPE) or WordPiece, which balance the size of the vocabulary and the coverage of rare words [34], [35]. The vocabulary also includes special symbols such as *[CLS]* or *[SEP]* and *[MASK]* (*[CLS]* is prepended to the sequence; its final contextual embedding is often used as a sequence-level summary for classification. *[SEP]* separates segments (A/B) or marks end-of-sequence).

Each token $w_i \in V$ is mapped to a d-dimensional vector via the *embedding matrix* $E \in \mathbb{R}^{|V| \times d}$:

$$x_i = E[w_i].$$

The matrix E is a trainable parameter: it is typically initialized randomly (e.g. Gaussian) or from a pretrained checkpoint, and then optimized jointly with the rest of the model by gradient descent. These dense vectors carry lexical information but, by themselves, have no notion of order.

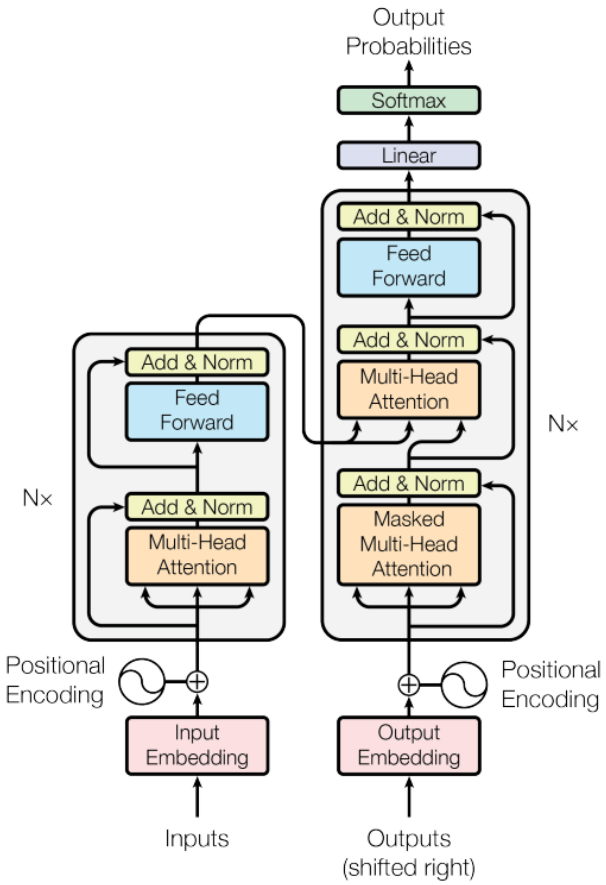


Figure 2.10: Transformer architecture from [33]. Left: encoder stack. Right: decoder stack.

Positional encodings

Transformers do not have any built-in sense of order: Without extra signals, a sequence of token vectors would look like a bag of points. To inject order, we *add* a position code for each token embedding. Let $x_i \in \mathbb{R}^d$ be the embedding of the i-th token and $\mathrm{PE}_{\mathrm{pos}(i)} \in \mathbb{R}^d$ its position code. The encoder input becomes

$$z_i = x_i + \mathrm{PE}_{\mathrm{pos}(i)}$$

This simple vector addition lets the model know both *what* a token is (via x_i) and *where* it is (via PE).

Fixed sinusoidal encodings [33] give each position a smooth, multi-frequency

signature:

$$\mathrm{PE}_{(pos,2j)} = \sin\Big(\frac{pos}{10000^{2j/d}}\Big), \qquad \mathrm{PE}_{(pos,2j+1)} = \cos\Big(\frac{pos}{10000^{2j/d}}\Big),$$

so nearby positions have similar phases and relative offsets are easy to infer with dot products. These codes extrapolate naturally to longer sequences.

Learned positional embeddings (used by BERT/RoBERTa) replace the sinusoid with a trainable table $P \in \mathbb{R}^{L\times d}$ and use

$$z_i \;=\; x_i \;+\; P[\mathrm{pos}(i)],$$

which adapts to the corpus/task but is tied to the maximum length L. In practice, many encoder stacks also sum a small *segment/type* vector to distinguish parts (A/B) of paired inputs.

Self-attention

Self-attention lets each token *look around* the whole sequence, score other tokens by relevance, and then average their information. It is a content-based, parallel alternative to recurrence.

Given position-aware inputs $Z \in \mathbb{R}^{n\times d}$, we form

$$Q = ZW^Q, \qquad K = ZW^K, \qquad V = ZW^V,$$

$$S \;=\; \frac{QK^\top}{\sqrt{d_k}}, \qquad A \;=\; \mathrm{softmax}(S{+}M), \qquad Y \;=\; AV.$$

where $Q, K \in \mathbb{R}^{n\times d_k}$ are the *queries/keys* and $V \in \mathbb{R}^{n\times d_v}$ are the *values*; the matrices W^Q, W^K, W^V are trainable projections with shapes $d\times d_k$, $d\times d_k$, and $d\times d_v$; $S \in \mathbb{R}^{n\times n}$ are *scaled dot-product* scores; the factor $1/\sqrt{d_k}$ keeps logits numerically stable; M is an optional mask (padding mask to ignore pads, or causal mask to block future positions in decoders); $A \in \mathbb{R}^{n\times n}$ are row-normalized attention weights; and $Y \in \mathbb{R}^{n\times d_v}$ contains the *contextualized* outputs with $y_i = \sum_j A_{ij} v_j$.

Self-attention gives each token a global receptive field in one step, so long-range dependencies are easy to model and training is fully parallel. Because weights A_{ij} depend on the content, the representation of a word adapts to the context (polysemy handling). Masks make the same mechanism usable for both encoders (padding) and

decoders (causality).

Multi-head attention

A single attention head offers one perspective on relevance. To let the model view context through multiple complementary lenses, the transformer uses *multi-head attention*: several heads run the same mechanism in parallel, each in its own subspace, and their outputs are combined. Building on single-head attention (previous subsection), we define for head $i = 1, \ldots, H$ with head size d_h (so $d = Hd_h$):

$$Q_i = Z\,W_i^Q,$$

$$K_i = Z\,W_i^K,$$

$$V_i = Z\,W_i^V,$$

$$h_i \;=\; \mathrm{softmax}\Big(\tfrac{Q_i K_i^\top}{\sqrt{d_h}} + M\Big)\,V_i,$$

$$Y \;=\; \mathrm{Concat}(h_1, \ldots, h_H)\,W^O.$$

where $Z \in \mathbb{R}^{n\times d}$ are position-aware inputs; $W_i^Q, W_i^K, W_i^V \in \mathbb{R}^{d\times d_h}$ are learnable per-head projections; $Q_i, K_i, V_i \in \mathbb{R}^{n\times d_h}$ are the queries, keys, and values for head i; M is an optional mask (padding or causal); $h_i \in \mathbb{R}^{n\times d_h}$ is the head output; Concat stacks heads along features giving an $n \times (Hd_h)$ tensor; and $W^O \in \mathbb{R}^{(Hd_h)\times d}$ projects back to the model dimension. The factor $1/\sqrt{d_h}$ plays the same stabilizing role as $1/\sqrt{d_k}$ in single-head attention.

Add & Norm and feed-forward.

Each transformer sublayer (attention or feed-forward) is wrapped by a *residual connection* plus *layer normalization*. Concretely, the output of the sublayer is *added* to its input (the residual path), and then *sum* is *normalized per token across features*. This keeps information flowing and stabilizes training. The idea of residual connections was popularized in computer vision by ResNet, where skip connections enabled very deep models [36]. We will return to ResNet in detail later.

The *feed-forward* part of a block is simply a small *MLP* applied *independently at each position*, with the *same* weights for all tokens. It adds nonlinearity and briefly expands the feature dimension before projecting back, enriching each token's representation without mixing positions (attention does the mixing).

Encoder and decoder

We now assemble the building blocks into the full transformer. The *encoder* turns a sequence of token+position embeddings into deep, contextual representations. The *decoder* generates an output sequence, using masked self-attention for causality and cross-attention to read from the encoder's representations.

Encoder Given token+position inputs $Z \in \mathbb{R}^{n \times d}$, the encoder stacks L identical blocks:

$$H^{(0)} = Z, \qquad H^{(\ell)} = \mathrm{Block}\big(H^{(\ell-1)}\big), \;\; \ell = 1, \ldots, L,$$

where each $\mathrm{Block}(\cdot)$ is MHA $\rightarrow$ Add&Norm $\rightarrow$ FFN $\rightarrow$ Add&Norm as described earlier. The top states $H^{(L)}$ are contextual token representations.

Decoder The decoder also stacks L blocks, but each block has *masked* self-attention for causality and *cross-attention* to the encoder output. Let $U^{(0)}$ be target embeddings with positions (shifted right by a start token). For layer ℓ:

$$U^{(\ell)} \;=\; \mathrm{DecBlock}\big(U^{(\ell-1)};\; H^{(L)}\big),$$

with the block defined compactly as

$$\mathrm{DecBlock}(U; H) \;=\; \mathrm{FFN}\Big(\mathrm{MHA}\big(Q = \underbrace{\mathrm{MHA}_{\mathrm{mask}}(U)}_{\text{causal self-attention}},\; K = H,\; V = H\big)\Big),$$

where $\mathrm{MHA}_{\mathrm{mask}}(\cdot)$ is the same multi-head attention as before but with a causal mask M that sets scores to $-\infty$ for future positions (so position t cannot attend to any $t' > t$). The cross-attention reuses the encoder's top states as keys/values ($K{=}V{=}H^{(L)}$) and the masked self-attention output as queries Q. As in the encoder, each sublayer is wrapped by Add&Norm (residual + layer norm), omitted here for brevity.

Encoder-only vs. decoder-only. For *representations* (classification, retrieval) one typically uses the encoder stack alone. For *generation* without a source sequence (language modeling), a decoder-only stack with causal masks is used. Tasks like translation benefit from the full encoder-decoder pair.

2.2.4 BERT, RoBERTa, and generative LLMs

BERT (encoder-only). BERT [37] is an encoder stack pretrained by an *unsupervised* objective that predicts masked tokens from bidirectional context (*masked language modeling*, MLM), plus the original *next sentence prediction* (NSP) task. Let $x_{1:n}$ be a tokenized input and M the set of masked positions. The MLM objective is

$$\max_{\theta} \sum_{m \in M} \log p_\theta\big(x_m \mid x_{\setminus M}\big),$$

i.e., predict each hidden token x_m given the remaining visible context. NSP classifies whether segment B follows segment A, but later work found it non-essential. After pretraining, BERT is fine-tuned by adding a small task head and updating all parameters.

Finally, we arrive at representations suitable for downstream use. As with BoW/T-FIDF, once text is mapped to a fixed-dimensional vector, standard supervised models (e.g. logistic regression, random forests, MLPs) can be trained in the usual way.

Given the final hidden states $H^{(L)} \in \mathbb{R}^{n \times d}$: (i) the special *[CLS]* vector h_{CLS} serves as a sequence-level summary; (ii) for token-level tasks (e.g. named-entity recognition) one reads per-token states $H^{(L)}_{i:}$; (iii) sentence/document embeddings can be obtained by mean pooling (or by mixing top layers). Pretraining is unsupervised; optionally attaching a small task head and fine-tuning yields supervised adaptation. Even without a head, the pooled vectors can already be used in many tasks.

RoBERTa (encoder-only, training refinements). RoBERTa [38] keeps the encoder architecture but drops NSP, uses *dynamic masking* (the mask pattern changes across epochs so the model sees more contexts), trains much longer, on more data, with larger batches and longer sequences. These changes substantially improve downstream results. Representations are extracted exactly as in BERT (e.g. *[CLS]* or pooled token states) and fine-tuned with a small task head when supervision is available.

Generative LLMs (decoder-only). Models like GPT, LLaMA, or ChatGPT employ the decoder stack with causal attention mechanism and are pretrained by *next-token prediction*:

$$\max_{\theta} \sum_{t=1}^{T} \log p_\theta\big(x_t \mid x_{<t}\big), \qquad p(x_{1:T}) = \prod_{t=1}^{T} p(x_t \mid x_{<t}).$$

Training is highly parallel within each step (masked attention lets all positions in a sequence train together), but *generation* at test time is inherently sequential: tokens

are produced one by one, each conditioned on the history. The *generative head* is a linear map to vocabulary logits followed by softmax (often tied to the input embedding matrix).

2.3 Image representations

We now turn to images. As with text, the goal is to map each image into a fixed-dimensional representation so that standard supervised models or end-to-end classifiers can be trained. This section briefly reviews the evolution of image representations, from handcrafted descriptors (SIFT, HOG, Bag-of-Visual-Words) to modern deep convolutional encoders, to position the representations studied in this dissertation. While classical descriptors are included as historical baselines, the focus of this thesis is on *deep representations learned by CNNs*, in particular ResNet-style encoders trained at large scale. In later chapters, we treat these encoder representations as the primary objects of analysis: we audit their interpretability and robustness using explanation-guided criteria and evaluate how targeted fine-tuning objectives can improve representation quality without degrading clean accuracy.

2.3.1 Classical descriptors: SIFT/HOG and Bag-of-Visual-Words

Before the advent of deep learning, image representations were built using engineered descriptors. Two of the most influential methods are *Scale-Invariant Feature Transform (SIFT)* [39] and *Histograms of Oriented Gradients (HOG)* [40].

SIFT SIFT detects salient keypoints stable across scale and rotation. For each keypoint p, compute image gradients in a local patch: $m(x,y) = \sqrt{g_x^2 + g_y^2}$ and $\theta(x,y) = \operatorname{atan2}(g_y, g_x)$. The patch is split into 4×4 spatial cells; in each cell, gradient magnitudes are accumulated into $B = 8$ orientation bins:

$$h_{c,b} = \sum_{(x,y)\in \text{cell } c} \mathbf{1}\big[\theta(x,y) \in \text{bin } b\big]\, m(x,y).$$

Concatenating all cell-bin values yields:

$$z = \big(h_{c,b}\big)_{c=1..16,\ b=1..8} \in \mathbb{R}^{128}.$$

HOG HOG describes an image window by dividing it into cells and accumulating histograms of gradient orientations. Let $g(x, y)$ be the gradient at (x, y) with orientation $\theta(x, y)$ and magnitude $\|g(x, y)\|$. The cell histogram is

$$h_b = \sum_{(x,y)\in\text{cell}} \mathbf{1}\big[\theta(x, y) \in \text{bin } b\big] \, \|g(x, y)\|.$$

Concatenating histograms from all cells and normalizing them within overlapping blocks yields a dense representation

$$z \in \mathbb{R}^d.$$

Bag-of-Visual-Words (BoVW) Local descriptors (e.g. SIFT or HOG) do not yield a fixed-length representation because the number of keypoints varies per image. The solution – analogous to text Bag-of-Words is to quantize descriptors into a *visual vocabulary*. A codebook $C = \{c_1, \ldots, c_K\}$ is learned by clustering descriptors (typically with k-means). Each descriptor $\phi(p)$ is assigned to its nearest centroid $j^\star = \arg\min_j \|\phi(p) - c_j\|_2$, and the image is represented by a histogram of visual-word counts:

$$h_j = \big|\{\, p : j^\star(p) = j \,\}\big|, \quad j = 1, \ldots, K; \qquad z = (h_1, \ldots, h_K) \in \mathbb{R}^K.$$

The resulting vector z is order-invariant and robust to small deformations, and can be used with a classifier such as an MLP. The pipeline of these "handcrafted" image representations is illustrated in Figure 2.11.

2.3.2 Convolutional Neural Networks (CNNs)

CNNs [41] learn features directly from pixels via *local connectivity* and *weight sharing*. A convolution layer applies a bank of small kernels to each input channel and aggregates the results, followed by a pointwise nonlinearity:

$$Y_c = \sigma\Big(\sum_k W_{c,k} * X_k + b_c \Big),$$

where $*$ denotes discrete convolution, k indexes input channels, c indexes output channels, and σ is typically ReLU. Weight-sharing drastically reduces the number of parameters versus fully connected layers and yields translation equivariance: shifting the input shifts the feature maps. Figure 2.12 illustrates the classical pipeline popular since

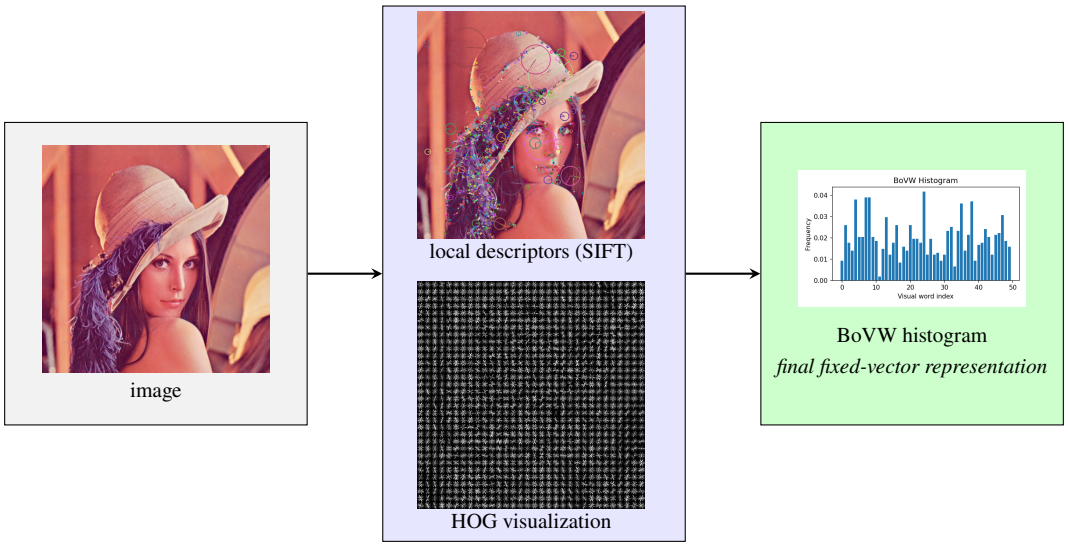


Figure 2.11: Pipeline of handcrafted image representations. The input image is transformed into local descriptors (e.g. SIFT or HOG). These are aggregated into a Bag-of-Visual-Words histogram, which constitutes a fixed-length representation usable by standard classifiers.

LeNet – a stack of convolutional layers (feature maps), interleaved with downsampling (subsampling/pooling), usually followed by a classification head. The early layers respond to edges and simple textures, the middle layers compose parts, and the deepest layers capture high-level structures. In practice, the final convolutional block produces a tensor that we flatten (or use Global Average Pooling) to obtain the representation z, which is then fed to a simple classification head (e.g. $y = \text{softmax}(WZ + B)$). Unlike order-invariant BoVW histograms that discard spatial layout, CNNs preserve structure and learn features jointly with the classifier; these features typically become truly discriminative only after end-to-end *supervised* training. However, the same convolutional encoder can be used in a convolutional autoencoder, allowing one to learn z without labels for fully unsupervised tasks.

2.3.3 Residual Networks (ResNet)

Compared to the plain CNN pipeline of Fig. 2.12 (stacked convolutions with downsampling followed by a classifier), ResNets add an *identity shortcut* on top of each convolutional block (Figure 2.13). This single change targets the *degradation problem* in deep plain CNNs (training error grows with depth) and makes very deep models both

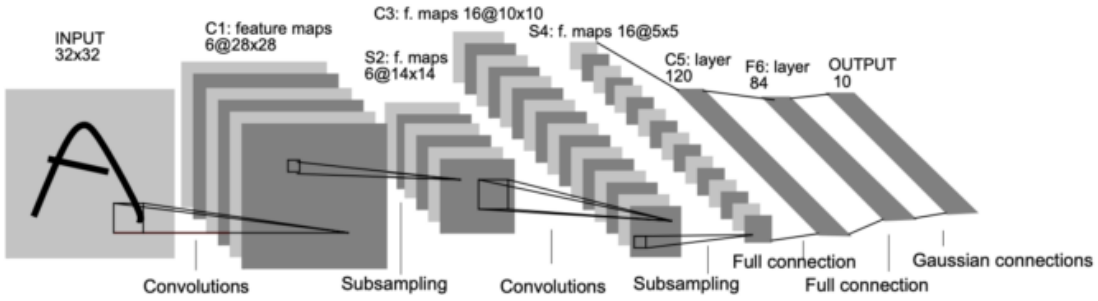


Figure 2.12: A classic CNN schematic (LeNet–5): input, stacks of convolutions with feature maps, subsampling/pooling, and a fully connected classifier. Adapted from Fig. 2 in [41]

trainable and accurate [36]. Concretely, a residual block learns the *residual* mapping $F(x) := H(x) - x$, where x is the input of the block and $H(x)$ the desired underlying transformation, and the results $y = F(x) + x$. If the desired mapping is close to identity, driving $F(x) \approx 0$ is easier than relearning $H(x) \approx x$. The skip path therefore creates a high-fidelity route for forward signals and backward gradients, stabilizing optimization at depths where plain CNNs struggle. When spatial size or channel count changes (e.g. via strided conv for downsampling), the shortcut is matched by a projection 1×1 W_s, which yields $y = F(x) + W_s x$. ImageNet-scale variants (ResNet-50/101/152) use a *bottleneck* $1\times1 \to 3\times3 \to 1\times1$ to first reduce and then restore width (channels), keeping compute in check while preserving accuracy [36]. A refinement, *preactivation*, moves normalization and activation before convolutions so that the bypass is a strict identity, further improving gradient flow [42].

In a plain CNN, we often flatten the last feature map (or apply global average pooling) and feed a small MLP; the resulting *pre-logits* vector is the image representation z. ResNets follow the same principle, but standardize on Global Average Pooling (GAP) after the final residual stage. Given the final feature tensor $H^{(L)} \in \mathbb{R}^{C\times H\times W}$ (channels $\times$ height $\times$ width), the fixed-length vector is

$$z_c = \frac{1}{HW} \sum_{u=1}^{H} \sum_{v=1}^{W} H^{(L)}_{c,u,v}, \qquad z \in \mathbb{R}^C,$$

which is passed to a linear+softmax head, $\hat{y} = \mathrm{softmax}(Wz + b)$, where W and b are classifier weights and bias, and $\hat{y}$ contains class probabilities. Relative to order-invariant BoVW, both CNNs and ResNets preserve spatial reasoning inside the encoder; relative to plain CNNs, residual shortcuts allow deeper hierarchies to form under the same

optimization budget, typically producing stronger features z after supervised end-to-end training. The same residual encoder can also be used without labels (e.g. autoencoders, contrastive pre-training), in which case z is taken from GAP while the learning objective differs from cross-entropy [36], [42].

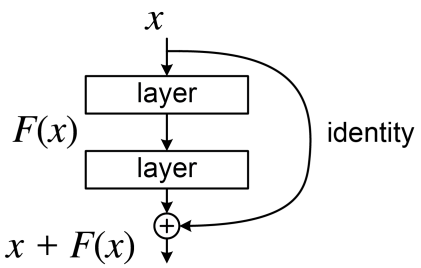


Figure 2.13: Residual block with identity shortcut: $y = F(x) + x$. Adapted from Fig. 2 in [36].

2.4 Clustering and Out-of-Distribution Detection for Auditing Representation Spaces

Vector representations $z = f_\theta(x) \in \mathbb{R}^d$, learned from tabular data, text, or images, encode the model's internal view of the data. Because many reliability properties depend directly on the structure of this space, methods that operate *purely on embeddings* provide a natural starting point for assessing representation quality.

In particular, *clustering* and *out-of-distribution (OOD) detection* can be applied without retraining the encoder and without access to task labels at inference time. Clustering probes whether the representation organizes samples into semantically coherent groups, while OOD detection tests whether the embedding space supports the separation of in-distribution data from novel or anomalous inputs. Together, these methods act as lightweight, representation-centric diagnostics that align directly with the audit stage of the *audit–measure–improve* workflow.

Unless stated otherwise, all methods below operate on a similarity or distance measure defined in the embedding space. We focus on two standard choices: Euclidean distance and cosine similarity,

$$\|z - z'\|_2 \quad \text{and} \quad \cos(z, z') = \frac{z^\top z'}{\|z\|_2 \, \|z'\|_2},$$

respectively. When an algorithm expects a dissimilarity measure, we use $1 - \cos(z, z')$ for cosine-based comparisons [43].

2.4.1 Clustering

Clustering is the task of grouping unlabeled points so that samples within a group are more similar to each other than to samples in other groups. Formally, given embeddings $Z = [z_1, \ldots, z_n]^\top$, clustering produces assignments $c_i \in \{1, \ldots, K\}$ (or marks points as *noise*), with the goal of high intra-cluster similarity and low inter-cluster similarity. Several clustering methods are outlined below. Illustrations that help build intuition are provided in Figure 2.14.

k-means. Given $Z = [z_1, \ldots, z_n]^\top \in \mathbb{R}^{n \times d}$ and a target number of clusters K, we seek centroids $\{\mu_k\}_{k=1}^K \subset \mathbb{R}^d$ and assignments $c_i \in \{1, \ldots, K\}$ that minimize the within-cluster sum of squares

$$\min_{\{\mu_k\}, \{c_i\}} \sum_{i=1}^{n} \left\| z_i - -\mu_{c_i} \right\|_2^2,$$

where z_i is the i-th embedding, n the number of samples, d the dimensionality, c_i the cluster index of z_i, and μ_k the centroid of cluster k. Let $C_k = \{i : c_i = k\}$. Note that K is a user-chosen hyperparameter specifying how many clusters/centroids we will attempt to fit and assign the data to. A classical solver is *Lloyd's alternating procedure*: [44].

1. **Init.** Choose $\mu_k^{(0)}$ for $k = 1, \ldots, K$ (e.g. k-means ++ seeding [45]).

2. **Repeat** for $t = 0, 1, 2, \ldots$ until reaching convergence:

   (a) *Assignment:*

   $$c_i^{(t+1)} = \arg \min_{k \in \{1, \ldots, K\}} \left\| z_i - -\mu_k^{(t)} \right\|_2^2.$$

   (b) *Update:* for each k with $C_k^{(t+1)} \neq \varnothing$,

   $$\mu_k^{(t+1)} = \frac{1}{|C_k^{(t+1)}|} \sum_{i \in C_k^{(t+1)}} z_i.$$

   If $C_k^{(t+1)} = \varnothing$, re-seed $\mu_k^{(t+1)}$ (e.g. farthest point).

3. **Stop** when assignments stop changing or when $\sum_{k=1}^K \|\mu_k^{(t+1)} - \mu_k^{(t)}\|_2 < \varepsilon$.

DBSCAN. Given embeddings $Z = [z_1, \ldots, z_n]^\top$ and a distance metric, DBSCAN uses two hyperparameters: neighborhood radius $\varepsilon > 0$ and minimum neighbors $minPts \in \mathbb{N}$ [46], [47]. For a point z, define its ε-neighborhood

$$\mathcal{N}_\varepsilon(z) = \{\, z_j \in Z : \; d(z_j, z) \leq \varepsilon \,\}.$$

A point is *core* if $|\mathcal{N}_\varepsilon(z)| \geq minPts$. A point is a *border point* if it is *not* core but lies within ε of some core point; border points are assigned to the cluster of that core. Point q is *directly density-reachable* from a core p if $q \in \mathcal{N}_\varepsilon(p)$. Point q is *density-reachable* from p if there exists a chain of directly density-reachable steps from p to q. Two points are *density-connected* if they are both density-reachable from some core o. A *cluster* is a maximal set of density-connected points; points not density-reachable from any core are labeled *noise*. The algorithm goes as follows:

1. **Init.** Mark all points as *unvisited*. Choose metric d (Euclidean by default; cosine is possible with a suitable ε).

2. **For** each unvisited point p:

   (a) Mark p as *visited*; compute $\mathcal{N}_\varepsilon(p)$.

   (b) *If* $|\mathcal{N}_\varepsilon(p)| < minPts$, label p as *noise* (may be reassigned later); **continue**.

   (c) *Else* start a new cluster C; add p and all points in $\mathcal{N}_\varepsilon(p)$ to a *seed set*.

   (d) **While** the seed set is non-empty:

      i. Pop q from the seed set. If q is *unvisited*, mark it *visited* and compute $\mathcal{N}_\varepsilon(q)$.

      ii. If $|\mathcal{N}_\varepsilon(q)| \geq minPts$, merge its neighbors into the seed set (cluster expansion).

      iii. If q is not yet assigned to any cluster, assign q to C.

Agglomerative clustering. Given embeddings $Z = [z_1, \ldots, z_n]^\top$ and a distance metric d, we start with each point as its own cluster and iteratively merge the closest pair according to a linkage rule until a stopping criterion is met. This bottom-up procedure yields a hierarchy of partitions [48], [49]. Common linkage rules (cluster-to-cluster dissimilarities) include:

$$\begin{aligned}
\text{single:} \quad & D(A,B) = \min_{a\in A,\, b\in B} d(a,b), \\
\text{complete:} \quad & D(A,B) = \max_{a\in A,\, b\in B} d(a,b), \\
\text{average:} \quad & D(A,B) = \frac{1}{|A||B|} \sum_{a\in A} \sum_{b\in B} d(a,b), \\
\text{Ward:} \quad & \Delta(A,B) = \frac{|A||B|}{|A|+|B|} \|\mu_A - \mu_B\|_2^2.
\end{aligned}$$

Here, $A, B \subseteq Z$ denotes clusters of sizes $|A|, |B|$ with centroids μ_A, μ_B. The algorithm goes as follows:

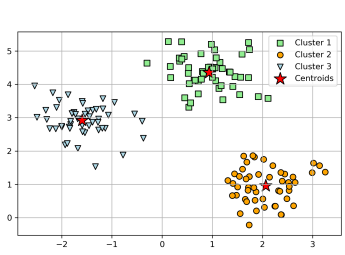
1. **Init.** Set clusters $\mathcal{C} = \{\{z_1\}, \ldots, \{z_n\}\}$. Precompute pairwise distances.
2. **Repeat** until the stopping condition holds:
   (a) Find the closest pair $(A^\star, B^\star) \subset \mathcal{C}$ using the chosen linkage.
   (b) Merge: $M = A^\star \cup B^\star$; update $\mathcal{C} \leftarrow (\mathcal{C} \setminus \{A^\star, B^\star\}) \cup \{M\}$.
   (c) Update $D(M, C)$ for all $C \in \mathcal{C}$ via the linkage formula.
3. **Output.** The dendrogram; cut at height h or choose K to obtain a flat clustering.

2.4.2 Evaluation of Clustering

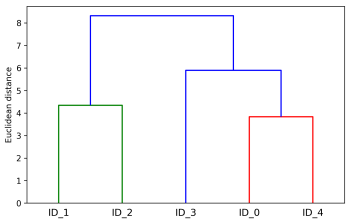
Clustering can in principle be viewed as an unsupervised classification problem: one may compare predicted cluster assignments against ground truth classes using confusion-matrix-based scores (e.g. accuracy, precision, recall). However, such measures assume that the number of predicted clusters matches the number of true classes and that the cluster labels can be aligned one-to-one with the ground truth labels. These assumptions rarely hold in practice. A more robust approach is to use criteria specifically designed for cluster evaluation, such as silhouette, ARI, or AMI.

Silhouette coefficient [51]. For point i, let a_i denote the mean distance to all points in its own cluster, and b_i the lowest mean distance to any other cluster. The silhouette is

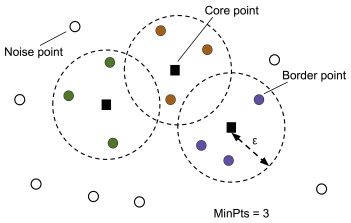
$$s_i = \frac{b_i - a_i}{\max(a_i, b_i)} \in [-1, 1],$$



k-means: Scatter with clearly marked centroids μ_k.



Agglomerative: A mini dendrogram illustrating the hierarchy.



DBSCAN: Schematic with ε – neighborhood circles and the distinction between core, border, and noise points.

Figure 2.14: Illustrations of three clustering families: centroid-based (k-means), hierarchical (agglomerative), and density-based (DBSCAN). Images come from [50].

and the overall score is the average of s_i. Higher values indicate that clusters are dense and well separated.

Adjusted Rand Index (ARI) [52]. Let $N = (n_{rs})$ be the contingency table between predicted clusters r and ground-truth classes s. The ARI is

$$\mathrm{ARI} = \frac{\sum_{r,s}\binom{n_{rs}}{2} - \frac{\sum_r \binom{n_{r\cdot}}{2}\sum_s \binom{n_{\cdot s}}{2}}{\binom{n}{2}}}{\frac{1}{2}\left[\sum_r \binom{n_{r\cdot}}{2} + \sum_s \binom{n_{\cdot s}}{2}\right] - \frac{\sum_r \binom{n_{r\cdot}}{2}\sum_s \binom{n_{\cdot s}}{2}}{\binom{n}{2}}}.$$

ARI looks at pairs of points: it counts how often the clustering and the ground truth agree (together vs. apart) and subtracts what would occur *by chance*. Hence, the score is equal to 1 for a perfect match, it is approximately 0 for random labels (after chance correction), and lower than 0 when the clustering is worse than random. The chance adjustment reduces bias when the number/size of clusters differs.

Adjusted Mutual Information (AMI) [53]. Let $N = (n_{uv})$ be the contingency table for predicted clusters U and true classes V, with row/column sums $n_{u\cdot}$ and $n_{\cdot v}$ and total $n = \sum_{u,v} n_{uv}$. Define empirical probabilities $p_{uv} = n_{uv}/n$, $p_u = n_{u\cdot}/n$, $p_v = n_{\cdot v}/n$. Then, with mutual information $\mathrm{MI}(U,V)$ and entropies $H(U), H(V)$ (any fixed log base), we

have

$$\mathrm{MI}(U,V) = \sum_u \sum_v p_{uv} \log \frac{p_{uv}}{p_u p_v} = \sum_u \sum_v \frac{n_{uv}}{n} \log\Big(\frac{n_{uv}\, n}{n_{u\cdot}\, n_{\cdot v}}\Big),$$

$$H(U) = -\sum_u p_u \log p_u, \qquad H(V) = -\sum_v p_v \log p_v,$$

$$\mathrm{AMI} = \frac{\mathrm{MI}(U,V) - \mathbb{E}[\mathrm{MI}]}{\max\{H(U), H(V)\} - \mathbb{E}[\mathrm{MI}]},$$

$$\mathbb{E}[\mathrm{MI}] = \sum_u \sum_v \sum_{t=t_{\min}}^{t_{\max}} \frac{t}{n} \log\Big(\frac{t\, n}{n_{u\cdot}\, n_{\cdot v}}\Big) \frac{\binom{n_{u\cdot}}{t}\binom{n-n_{u\cdot}}{n_{\cdot v}-t}}{\binom{n}{n_{\cdot v}}},$$

$$t_{\min} = \max\big(0,\, n_{u\cdot} + n_{\cdot v} - n\big), \qquad t_{\max} = \min\big(n_{u\cdot},\, n_{\cdot v}\big).$$

Adjusted Mutual Information (AMI) is bounded in $[0, 1]$ under this normalization (with $\max\{H(U), H(V)\}$). It measures how much information about the true labels is recovered by the clustering, beyond what is expected *by chance*. Thus, the score equals 1 when the partitions coincide up to a permutation, is approximately 0 for independent/random partitions, and remains comparable across different numbers of clusters due to chance correction and entropy normalization.

In practice, AMI is information-theoretic, while ARI is pairwise/combinatorial, and using both is beneficial. AMI normalizes by entropy, so it remains comparable across different numbers of clusters and is less biased when k changes or classes are highly imbalanced; it quantifies how much label information the clustering recovers (beyond chance). ARI, by contrast, scores pairwise consistency (together vs. apart) and is more sensitive to over/under-merging and boundary mistakes, providing a complementary view of partition quality. Because they emphasize different failure modes (AMI: information overlap and comparability across k; ARI: pairwise agreement and sensitivity to granular errors), reporting both yields a more reliable, permutation-invariant assessment. When ground truth is unavailable, the Silhouette complements them by evaluating cohesion and separation purely from the data geometry.

2.4.3 Out-of-Distribution (OOD) Detection

In many applications, models are deployed in environments where inputs may differ significantly from the training data. Detecting such *out-of-distribution* (OOD) samples is crucial for ensuring reliability and safety: an OOD point should ideally be flagged rather than classified with high confidence into one of the known classes. Failure to

detect OOD inputs may lead to misleading predictions, spurious confidence estimates, or even security vulnerabilities [15].

Formally, given embeddings of test points z and a reference set of *in-distribution* (ID) embeddings, the goal is to compute a score function $s(z)$ such that OOD samples can be separated from ID ones. A sample is declared OOD if the score crosses a threshold, i.e., $s(z) < \tau$ (or $s(z) > \tau$), with τ chosen on a validation split.

Clustering methods can be used for OOD detection. For example, DBSCAN labels sparse regions as noise, which can be interpreted as OOD. Similarly, in centroid-based methods, points lying far from all cluster centers may be rejected. This highlights a close connection between clustering and novelty detection: both rely on defining regions of high density or similarity as ID and flagging points outside those regions as OOD. OOD approaches can be broadly categorized into:

- **Parametric methods**, which assume a specific distributional form for the embeddings (e.g. multivariate Gaussian). Parameters such as mean and covariance are estimated from training data, and test points are scored via likelihood or Mahalanobis distance.

- **Non-parametric methods**, which do not posit an explicit distribution but rely on the geometry of the reference set (e.g. nearest-neighbor distances). These methods adapt naturally to complex distributions, but may be less scalable.

In what follows, we describe the specific parametric and non-parametric methods used in this work.

kNN distance (non-parametric). Define the OOD score as the negative distance to the k-th nearest ID neighbor (or the mean of the first k),

$$s_{\mathrm{kNN}}(z) \;=\; -\,\|z - \mathrm{NN}_k(z; Z_{\mathrm{ID}})\|_2,$$

so large distances imply isolation (anomaly).

Mahalanobis distance to class Gaussians (parametric). Assuming $z \mid y = c \sim \mathcal{N}(\mu_c, \Sigma)$ with tied covariance, score OOD by the negative minimum Mahalanobis distance [54]:

$$s_{\mathrm{Mahalanobis}}(z) \;=\; -\min_c\,(z-\mu_c)^\top \Sigma^{-1}(z-\mu_c).$$

In practice, ℓ_2-normalizing features often helps both methods; whitening can further stabilize distances. kNN is label-free and multi-modal; Mahalanobis is strong when class means/covariance are well estimated.

2.4.4 Evaluation of OOD Detection

OOD detection can be naturally cast as a binary classification problem: each test sample is either in-distribution (ID) or out-of-distribution (OOD). Consequently, the same evaluation metrics (previously introduced) as in standard classification apply, such as precision, recall, F1, AUROC, and FPR@95%. Among these, AUROC summarizes the overall ranking quality, while FPR@95% is widely used in previous work as a concrete operating-point measure. In practice, reporting both provides a balanced view: AUROC reflects separability across thresholds, and FPR@95% quantifies performance at high recall for ID.

2.5 Explainability

The term Explainable Artificial Intelligence (XAI) refers to methods and techniques that make the behavior and decisions of AI systems understandable to humans. According to a comprehensive definition, XAI aims to produce models that not only perform well, but also provide interpretable and transparent explanations of their predictions or internal mechanisms to human users.

A compelling argument for the necessity of explainability in machine learning models is illustrated in [55], where the authors refer to the famous “Clever Hans” phenomenon. Clever Hans was a horse believed to perform basic arithmetic by tapping his hoof. However, it was later revealed that the horse was not actually solving the problems but responding to involuntary cues from his owner.

This anecdote is analogous to situations in machine learning where a model appears to perform well, yet it bases its predictions on spurious correlations or irrelevant features. Several studies have demonstrated such behavior. For example, in the case of prediction of the risk of pneumonia, a model learned that patients with asthma had a lower risk, not because asthma is protective, but because these patients received more intensive care [56]. Similarly, in computer vision tasks, classifiers have been shown to rely on background textures or watermarks instead of the actual object of interest [57].

Without knowing what features the model is focusing on, it becomes difficult to

assess:

1. **Transparency**, that is, the ability to understand how the model arrives at its decisions, and

2. **Trustworthiness**, that is, the confidence that one can place in the model's output.

Although transparency and trust may not be essential for all applications (e.g. generative models for images), they become critical in safety-sensitive contexts. As artificial intelligence becomes increasingly embedded in critical systems, explainability becomes a necessary component of responsible AI deployment.

Consider the example of autonomous vehicles, where understanding the reasoning behind a car's decision in real time could be the difference between safety and disaster. Similarly, AI-powered identity verification systems, network threat detection, or medical diagnostic tools must be able to explain not only legal or ethical compliance but also improve the robustness of the model and mitigate bias.

Explanations also enable human users to audit and improve models. Interpretability can provide novel insights into the data or the problem domain itself, beyond what the raw accuracy metrics reveal. In addition, presenting explanations to experts in the field might improve their efficiency. These users may not only rely on the classifier's decision, but also need to understand the key features that contributed to it. This allows human-in-the-loop decision-making, which is often vital in high-stakes environments.

This is precisely why explainability plays such a crucial role in the context of this work. It enables the preservation of the trustworthiness of a model, which is essential from the point of view of the security of the information system. Moreover, explainability can help reveal a model's limitations or vulnerabilities, which can then be exploited, either for model improvement or, in adversarial contexts, for malicious purposes.

In this section, we present a taxonomy of explainability methods, adapted in part from [58], to provide a structured overview of existing approaches.

Taxonomy of Explainability Methods Explainability methods can be categorized along several axes:

- **Based on the data type:**
  - Tabular data

  - Images
  - Text

- **Based on the number of samples:**
  - *Local explainability* - explains the model's decision for a single, specific instance.
  - *Global explainability* - provides a general understanding of the model's behavior, or an aggregated explanation over a dataset.

- **Based on the relationship with the model:**
  - *Model-agnostic methods* - methods that can be applied to any type of model, regardless of its internal structure.
  - *Model-specific methods* - methods designed for a particular class of models (e.g. decision trees, neural networks).

- **Based on the purpose of explainability:**
  - Explaining black-box model predictions
  - Improving model performance and reliability
  - Testing sensitivity of predictions to perturbations
  - Attacking or probing the model for weaknesses

2.5.1 Inherently Interpretable Models

Inherently interpretable models are those whose structure and parameters allow human experts to directly understand how input features affect the output. Such models serve as a foundation for surrogate explanations in methods like LIME and SHAP, described later. Common examples include linear regression, logistic regression, decision trees, and generalized additive models (GAMs).

Linear Regression. In a linear regression model, the relationship between an input vector $x = [x_1, \ldots, x_n]$ and the continuous output $\hat{y}$ is given by

$$\hat{y} = \beta_0 + \beta_1 x_1 + \cdots + \beta_n x_n.$$

Each coefficient β_i directly reflects the contribution of the corresponding feature x_i, making the model straightforward to interpret.

Logistic Regression. For binary classification tasks, logistic regression models the probability of a positive outcome using the logistic function:

$$P(y = 1 \mid x) = \sigma\Big(\beta_0 + \beta_1 x_1 + \cdots + \beta_n x_n\Big),$$

where the sigmoid function is defined as

$$\sigma(z) = \frac{1}{1 + e^{-z}}.$$

The coefficients β_i (or rather $\exp \beta_i$) again offer direct insight into the influence of each feature, thus providing a transparent explanation of the prediction.

Decision Trees. Decision trees partition the feature space using a hierarchy of if-then-else rules. Each internal node in the tree represents a decision based on a feature threshold, while each leaf node corresponds to a predicted output. This structure allows one to trace the prediction path from the root to the leaf, clearly illustrating the sequence of decisions that lead to the final prediction.

Generalized Additive Models (GAMs). GAMs generalize linear models by allowing for non-linear relationships via smooth functions f_i of each individual feature:

$$\hat{y} = \beta_0 + f_1(x_1) + f_2(x_2) + \cdots + f_n(x_n).$$

Because each function f_i depends only on one feature, GAMs strike a balance between capturing non-linear effects and maintaining interpretability.

2.5.2 Perturbation-based Explanation Methods

One of the fundamental categories of explainability techniques consists of perturbation-based methods, which analyze how local changes in the input affect the model's predictions. These approaches typically treat the model as a black box and observe its behavior in response to systematically modified input features.

A classical technique in this group is *Sensitivity Analysis (SA)*, which computes the gradient of the model output with respect to the input. While historically used for model interpretation, SA does not directly explain the prediction itself but rather how it changes

with slight input variation. Moreover, it suffers from several shortcomings, including *gradient shattering* and explanation discontinuities, which limit its effectiveness for modern deep models [59].

Several extensions have been proposed to address these limitations. For instance, *SmoothGrad* [60] improves SA by averaging gradients over multiple noisy versions of the input, which helps reduce instabilities. Similarly, *Integrated Gradients* [61] compute the path-integrated gradient between a baseline and the actual input, yielding more consistent and theoretically grounded attributions.

More general perturbation-based methods go beyond gradients and rely on explicit input modification. A well-known example is the *occlusion method* [62], which masks parts of the input (e.g. patches of an image) and evaluates how the prediction changes. The *Prediction Difference Analysis (PDA)* approach [63] uses conditional sampling to replace parts of the input while maintaining data distribution, effectively removing information while preserving structure. Both methods are model-agnostic and applicable to any classifier but tend to be computationally intensive due to the repeated forward passes required.

An advanced variant in this family is the *Meaningful Perturbation* method [64], which frames explanation as an optimization problem. The objective is to find the smallest possible change to the input that causes a significant drop in the model's confidence. This results in saliency maps that highlight the most informative regions for a given prediction.

Conversely, *Activation Maximization* [65] takes the opposite approach: instead of minimizing the output, it synthesizes inputs that maximize the activation of specific units in the model (e.g. neurons or class scores). These inputs serve as prototypes of the learned concepts and can be used to understand what the model has internalized. Although conceptually related to meaningful perturbation, activation maximization focuses on representative patterns rather than prediction-specific explanations.

While perturbation-based techniques can be computationally expensive, they offer valuable insights, especially in black-box scenarios. As such, they remain a core component of modern explainability research.

2.5.3 Explaining with Surrogate Models

Another major category of explainability methods relies on surrogate models, which aim to approximate the behavior of a complex, less interpretable model using a simpler,

inherently interpretable one. These methods can yield either local or global explanations depending on the surrogate's scope.

A prominent example is *LIME (Local Interpretable Model-agnostic Explanations)* [66]. LIME explains the prediction for a specific input instance by training a simple model locally around the input. In LIME, the surrogate model is typically chosen to be a linear model of the form

$$\hat{g}(z) = \phi_0 + \sum_{j=1}^{J} \phi_j z_j,$$

where z represents an interpretable (often binary) version of the original input, and ϕ_j are the corresponding coefficients indicating feature importance. The surrogate is obtained by minimizing an objective of the form

$$\mathcal{L}(f, g, \pi_x) + \Omega(g),$$

where $\mathcal{L}(f, g, \pi_x)$ measures the fidelity of the surrogate g to the original model f in the locality defined by the proximity kernel π_x, and $\Omega(g)$ is a complexity penalty that ensures g remains simple and interpretable.

To ensure interpretability, surrogate models are intentionally restricted to simple, explainable forms. A common choice is *logistic regression*. In a logistic regression model, the probability of a positive outcome is given by

$$P(y = 1 \mid x) = \sigma\Big(\beta_0 + \sum_{j=1}^{J} \beta_j x_j\Big),$$

with the logistic function defined as $\sigma(z) = \frac{1}{1+e^{-z}}$. The model's coefficients β_j provide clear insight into the contribution of each feature, serving as an inherently interpretable explanation.

Another highly influential method in this category is *SHAP (SHapley Additive Explanations)* [67]. SHAP unifies several existing explanation techniques under the framework of cooperative game theory. It attributes importance scores to input features based on their marginal contribution to the prediction, averaged over all possible feature subsets. The Shapley value for feature i is given by

$$\phi_i = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(|N| - |S| - 1)!}{|N|!} \Big[f_{S\cup\{i\}}(x_{S\cup\{i\}}) - -f_S(x_S) \Big],$$

where N is the set of all features and $f_S(x_S)$ denotes the model's prediction when only the features in subset S are present. This formulation ensures desirable properties such as consistency and local accuracy.

Building on LIME, *Anchors* [68] extend this idea by providing rule-based explanations. Anchors identify a set of conditions that are sufficient to guarantee a specific prediction with high precision, yielding human-readable if-then rules that are particularly useful with categorical or structured data.

Surrogate-based explanations offer a flexible, model-agnostic toolkit for interpreting predictions. However, their fidelity to the original model must be carefully evaluated to ensure reliable insights.

Application to Different Data Modalities: When applying surrogate models across various data types, some adaptations are necessary:

- **Tabular Data:** Surrogate methods like LIME and SHAP operate directly on feature vectors. The inherent interpretability of tabular data, where each column represents a distinct variable, allows simple models such as logistic regression to effectively capture and explain feature impacts.

- **Text Data:** For textual data, surrogate models typically rely on bag-of-words or embedding representations. In this context, LIME/SHAP may generate perturbed texts by removing or modifying words, and the resulting local model explains the importance of specific terms in the classification.

- **Image Data:** In the case of images, surrogate models are applied to segments or superpixels rather than raw pixels. Techniques like the occlusion method or SHAP on image regions help to identify the most salient areas that contribute to a prediction.

2.5.4 Model-Specific Approaches: Leveraging Structure, Backpropagation, and Attention Mechanisms

Model-specific explainability methods utilize the internal structure of a model to provide detailed insights into its decision-making process. These approaches, which are tailored to specific architectures such as deep neural networks and transformers, rely on access to gradients, activations, and attention weights.

Gradient-Based Methods. A common family of techniques are gradient-based methods. For example, *Integrated Gradients* computes the path-integrated gradient from a baseline x' to the actual input x to attribute the prediction to each feature:

$$\text{IntegratedGradients}_i(x) = (x_i - x_i') \int_{\alpha=0}^{1} \frac{\partial f(x' + \alpha(x - x'))}{\partial x_i} \, d\alpha.$$

This method leverages gradient information to ensure that attributions satisfy desirable axiomatic properties.

Layer-wise Relevance Propagation (LRP). LRP [69] propagates the prediction score backward through the network by redistributing relevance layer-by-layer. The relevance R_i for neuron i is computed as:

$$R_i = \sum_j \frac{a_i w_{ij}}{\sum_{i'} a_{i'} w_{i'j} + \epsilon} R_j,$$

where a_i is the activation of neuron i, w_{ij} is the weight from neuron i to neuron j, and ϵ is a small constant to avoid division by zero.

DeepLIFT. *DeepLIFT* [70] addresses the shortcomings of standard gradient methods by comparing each neuron's activation to its reference activation. It assigns contribution scores based on the differences:

$$C_{\Delta x} = \frac{\Delta y}{\Delta x},$$

where Δy is the difference in output between the actual input and a reference, and Δx represents the corresponding difference in input.

Grad-CAM. For convolutional neural networks, *Grad-CAM* [71] generates coarse localization maps by combining gradients with the feature maps of the last convolutional layer:

$$L^c_{\text{Grad-CAM}} = \text{ReLU}\left(\sum_k \alpha_k^c A^k \right),$$

with A^k representing the kth feature map and α_k^c computed as the global average of the gradients over A^k.

Transformer-Based Approaches and Attention Rollout. Transformers have gained popularity due to their self-attention mechanisms. While raw attention weights provide partial insight, methods like *Attention Rollout* [72] aggregate these weights over multiple

layers to capture the full information flow. For each transformer layer $l \in \{1, \ldots, L\}$, let $A^{(l)} \in \mathbb{R}^{n \times n}$ be the attention matrix, where n is the number of tokens. To account for residual connections, each matrix is augmented with the identity matrix:

$$R^{(l)} = A^{(l)} + I.$$

The overall attention is then given by the product of these augmented matrices:

$$A_{\text{rollout}} = \prod_{l=1}^{L} R^{(l)}.$$

The entry (i, j) in A_{rollout} quantifies the cumulative influence of token i on token j through the network, offering an intuitive visualization of the model's reasoning.

2.5.5 Visualization Techniques as Explainability Methods

Although often regarded as exploratory data analysis tools, visualization techniques can also serve as powerful methods for explainability by revealing the underlying structure of learned representations. In particular, methods such as Principal Component Analysis (PCA), t-distributed Stochastic Neighbor Embedding (t-SNE), and Uniform Manifold Approximation and Projection (UMAP) are widely used to project high-dimensional data into two or three dimensions for qualitative assessment.

Principal Component Analysis (PCA). PCA is a linear dimensionality reduction technique that projects data onto a lower-dimensional subspace. Given a mean-centered data matrix X, the covariance matrix is computed as

$$S = \frac{1}{n-1} X^\top X,$$

where n is the number of samples. The eigen decomposition of S provides the eigenvectors and eigenvalues, and the data is projected onto the eigenvectors corresponding to the largest eigenvalues. This projection often reveals the most dominant variance patterns in the data and serves as an intuitive explanation of model behavior [73].

t-Distributed Stochastic Neighbor Embedding (t-SNE). t-SNE is a nonlinear technique that preserves local structure by converting high-dimensional distances into conditional probabilities. It minimizes the Kullback-Leibler divergence between the

probability distributions in the original and embedded spaces:

$$\mathrm{KL}(P\|Q) = \sum_{i \neq j} p_{ij} \log \frac{p_{ij}}{q_{ij}},$$

where p_{ij} and q_{ij} represent the pairwise similarities in the high-dimensional and low-dimensional spaces, respectively [74]. t-SNE is particularly effective at visualizing clusters, which can indicate how well a model separates different classes.

Uniform Manifold Approximation and Projection (UMAP). UMAP is a more recent nonlinear dimensionality reduction method that leverages concepts from manifold theory and fuzzy topological representations. It constructs a weighted graph of high-dimensional data and optimizes a low-dimensional embedding by minimizing the loss of cross entropy between fuzzy simplicity sets in both spaces [75]. UMAP often preserves both local and global data structure, making it valuable for understanding the overall distribution and the learned representations.

Visual explanations derived from these techniques allow researchers to inspect and interpret the internal representations of complex models. For example, if the features learned by a neural network yield well-separated clusters in a t-SNE or UMAP plot, this can be seen as evidence of the discriminative capacity of the model. In contrast, overlapping clusters can indicate ambiguity or bias in the model representations. Visual explanations provide a compelling means to assess model performance and identify potential areas for improvement, complementing more formal explainability methods.

2.6 Robustness of Machine Learning

Robustness is the ability of a model to maintain reliable performance when conditions deviate from the idealized training setup. Deviations include *benign* perturbations (noise, blur, missing values), *distribution shifts* (new subpopulations, changed acquisition conditions), and *adversarial* manipulations crafted to cause errors. A robust system should keep accuracy and *calibration* under control across these changes, avoid overconfidence on unfamiliar inputs, and resist small but harmful perturbations [76], [77].

A common evaluation protocol uses stress-test suites that simulate real-world degradations. *Corruption robustness* is assessed with benchmark families such as CIFAR-*-C and ImageNet-C/-P (noise, blur, weather, digital artifacts, and perturbation

sequences), reporting corruption error and stability [77]. *Distribution shift* is measured in matched-but-new test sets (e.g. ImageNet-V2) or in curated domain-shift collections such as WILDS (wildlife cameras, satellite, medical, etc.), emphasizing subpopulation robustness and generalization [78], [79]. Many failures trace back to *spurious correlations*: models exploit shortcuts (e.g. texture bias, backgrounds) that break under shift [80]. Complementary diagnostics include calibration metrics, confidence risk curves, OOD detection (Sect. 3.3.2) and ablations with missing or perturbed features [76].

Models may (i) degrade under small input corruptions or slight changes in acquisition conditions, (ii) misgeneralize to new subpopulations, (iii) remain overconfident on OOD inputs, and (iv) be vulnerable to adversarial examples – small, often imperceptible changes that flip predictions [13]. These failures motivate stress-tested evaluation and defenses (briefly noted below) alongside our OOD and clustering analyzes. All of the above points will be addressed in this work, but at least one, adversarial attacks, requires a deeper introduction.

2.6.1 Adversarial Attacks: threat models and examples

An *adversarial example* is an input x' close to a benign x (under a constraint such as $\|x' - x\|_p \leq \varepsilon$) that causes a model to err. Attacks can be *untargeted* (any wrong label) or *targeted* (force a specific label). Capabilities range from *white-box* (full access to model and gradients) to *black-box* (only queries or transfer from surrogate models). Perturbation models include L_∞, L_2, L_0 budgets, but also domain-specific constraints (discrete tokens in text, monotone or box constraints in tabular). In physical settings, robustness is tested under transformations (*Expectation over Transformation*, EOT) [81]. A practical concern is *transferability*: examples crafted for one model often fool others, enabling black-box attacks [13].

Images (continuous domains). Canonical gradient-based methods include FGSM (single-step) and PGD (iterative, strong first-order baseline) [14]. Optimization-based attacks like Carlini-Wagner (C&W) and minimal-norm methods like DeepFool reduce perturbation size [82]. AutoAttack ensembles robust, parameter-free components for reliable evaluation [83]. Physical-world variants use adversarial patches or printed patterns that survive viewpoint changes [84].

Text (discrete, semantic constraints). Edits must preserve grammar and meaning. *HotFlip* applies gradient-guided character/word flips [85]. *TextFooler* searches for

synonym substitutions under semantic similarity and part-of-speech constraints. *BERT-Attack* leverages masked-LM proposals for fluent adversarial paraphrases [86]. Genetic and search-based methods operate with black-box objectives while enforcing semantic consistency.

Tabular data (constraints and validity). Attacks must respect feature ranges, discreteness, and sometimes monotonic or causal constraints. Gradient-based evasion with box and integer constraints produces effective perturbations in DNNs; for tree ensembles, dedicated optimization shows that targeted evasion is feasible [87]. In security-like settings (e.g. malware), attacks respect functionality constraints while altering sparse binary features [88].

Defenses (brief context). The most reliable empirical defense is *adversarial training*: min-max optimization with PGD examples [89]. TRADES balances robustness and accuracy via a KL penalty [90]. For certified guarantees, *randomized smoothing* turns a base classifier into one with L_2-certified radii [91]. Caution is warranted: many defenses fail under stronger evaluations or gradient issues [81]. Robustness can also trade off with standard accuracy on natural data [92].

2.7 Active Learning and Human-in-the-Loop

In the audit–measure–improve framework presented in this dissertation, human involvement plays a dual role. Human judgment can support qualitative assessment during auditing, and it can also be actively integrated into the *improve* stage through targeted supervision. This section introduces active learning and human-in-the-loop learning as the conceptual and methodological basis for the guided improvement strategies employed in our empirical studies.

A central problem in machine learning practice is how to improve the performance of a model on a given task while also guiding its behavior in a purposeful way. In many domains, vast amounts of unlabeled data are available, but only a small portion can be annotated due to cost or time constraints. The question then arises: *which data points should be labeled to maximize the improvement of the model?* Active learning (AL) provides a principled answer by letting the model itself select the most informative examples to query for labels [93].

Rather than labeling data at random, the learner actively identifies points where additional supervision would be most valuable, for example, instances about which it is

uncertain or which best reduce ambiguity in decision boundaries [94]. This strategy can substantially reduce annotation effort while maintaining or even improving accuracy. Importantly, by focusing annotation on critical or difficult cases, active learning also helps preserve precision, improve robustness, and mitigate the risk of models relying on spurious correlations. In this sense, active learning represents a shift from passive to guided model training.

The idea connects directly to the broader paradigm of *human-in-the-loop* machine learning, in which human expertise and algorithmic inference form a continuous feedback cycle [95]. Instead of relegating experts to a one-off labeling stage, these systems engage them dynamically: the model escalates uncertain or high-stakes cases for human judgment, while routine decisions remain automated. Examples include medical diagnostic assistants, where doctors validate ambiguous predictions, or content moderation pipelines that combine automated filters with human review.

Early studies have already demonstrated the effectiveness of active learning in domains such as text classification [94], and subsequent work extended it to computer vision and natural language processing [93]. Despite differences in setting, the underlying principle remains consistent: by guiding data collection and supervision, active learning and human-in-the-loop approaches help build models that are not only more accurate but also more efficient, robust, and trustworthy.

Chapter 3

Audit-Measure-Improve Illustrated on HTTP/URL Classification

To rigorously discuss the *trustworthiness* of learned representations and classifiers, we introduce the *audit–measure–improve* framework and illustrate its operation on a concrete, real-world problem. Specifically, we instantiate the proposed framework in a text-based security setting, studying the task of classifying HTTP headers and URLs.

This study provides a controlled yet practically relevant setting in which the full audit–measure–improve loop can be exercised: representations can be inspected for spurious cues, quantitatively evaluated using clustering and OOD-based measures, and iteratively refined through targeted interventions. While the application domain is security-oriented, the proposed methodology is generic and transferable to other modalities and tasks.

Research on HTTP/URL detection has progressed along three lanes: document embeddings, hand-crafted features, and supervised deep models. One stream uses document embeddings, such as Doc2Vec paired with ensembles, often aggregating ten requests into a single "document" and training on CSIC2010 as a whole [96]. Another line blends TFIDF weighting with Word2Vec and gradient boosting, reporting strong results on CSIC2010, UNSW-NB15, and MALICIOUSURL [97]. Representation learning with autoencoders over ASCII bigram features and anomaly scoring with Isolation Forest has also been explored [98]. URL-centric benchmarks based on lexical features compare classic learners on ISCXURL2016 [99], while feature selection or compact nine-feature recipes can nearly saturate scores [100], [101]. Purely supervised deep pipelines likewise

report excellent accuracy, including LSTM-CNN stacks and RNN/CNN ensembles built on traffic-specific tokenizers [102], [103]. Across these threads, interpretability and concept drift are rarely examined (to say nothing of the trustworthiness and robustness of the proposed algorithms). Choices like heavy aggregation or dataset quirks can inflate metrics, a concern echoed by analyses of benchmark "design smells" in Intrusion Detection Systems corpora [104]. We addressed some of these gaps in our earlier work [105], [106] by (i) proposing *Sec2vec*, a RoBERTa-based vectorization (probably the first work to propose using transformers in the given context) compared with BoW and fastText, evaluated via a downstream Random Forest; (ii) validating across datasets to probe generalization under concept drift; and (iii) using SHAP to surface human-readable token patterns that explain anomalies. These provide a solid starting point for the present chapter, which further (iv) evaluates clustering, OOD detection, and OOD generalization; (v) improves the model via fine-tuning; (vi) reassesses our metric portfolio on the improved model; and (vii) refines visualization and explainability. Together, these steps realize the *audit-measure-improve* loop. Each step is documented with concrete guidelines for building reliable and trustworthy representations.

3.1 Experimental Setup and Datasets

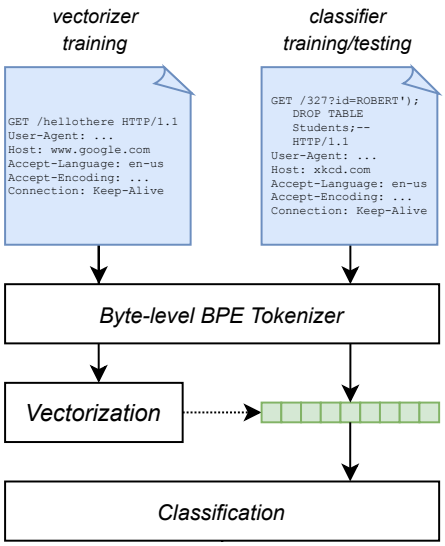


Figure 3.1: Our Sec2vec framework proposed in [106]

Figure 3.1 shows the framework proposed in our prior work [106], which serves as the starting point for this chapter. It maps raw HTTP headers and URLs (documents) to vector representations and then to downstream classifiers. In what follows, we reuse this pipeline to study how representations are learned, evaluated, and improved. We now describe each step of the framework.

Tokenization Byte Pair Encoding (BPE) was originally introduced as a data compression technique, where the most frequent pairs of bytes are replaced with a new symbol. Later it was adapted to natural language processing as a method of text tokenization [107], grouping the most frequent character sequences in the training corpus. The procedure begins with single characters as tokens and repeatedly merges frequent pairs into longer units until a vocabulary is formed. A widely used extension is Byte-level Byte Pair Encoding (BBPE), which operates on raw bytes rather than characters [108]. This keeps the vocabulary compact while enabling coverage of diverse forms.

For the tasks in this work, we employ a BBPE tokenizer trained on the entire dataset, including both normal and anomalous traffic. While training only on the subset later used for RoBERTa would not affect quantitative results, whole-corpus training avoids inconsistent token splits that may hinder interpretability in our later explainability analysis. The resulting tokens are subsequently used for all vectorization methods described in the following section.

Vectorization As vectorization methods we employ *Bag-of-Words (BoW)* and *RoBERTa*, all of which have been introduced in Chapter 2. These approaches provide complementary perspectives: BoW as a sparse lexical baseline and RoBERTa as a transformer-based model capable of capturing contextual dependencies. In the BoW representation, the last position of the vector corresponds to the out-of-vocabulary (OOV) bucket, which collects all tokens unseen during training. This design often allows the model to detect more anomalies, partially compensating for the inherent limitations of BoW. The resulting representations are later used for downstream classification and explainability analyses.

Classification Any suitable algorithm can be applied to classify the obtained vectors. In this work, we use the *Random Forest*. Although not reported in detail, we also tested a Multi-layer Perceptron (MLP) in every *presented* case (but not in every conceivable experiment), obtaining nearly indistinguishable results. Our choice to focus on a tree-based classifier is motivated by practicality: the time required for subsequent explainability analysis is significantly shorter compared to the MLP, which makes the

Random Forest a more efficient baseline for our study. It should also be emphasized that the purpose of this work is not to compete for the highest possible accuracy, but rather to investigate the trustworthiness and interpretability of learned representations.

3.1.1 Datasets

Table 3.1: Dataset divisions. The "Train" part is used to train the vectorizer (e.g. RoBERTa). The "Test" part is used for training and evaluating the classifier; this partition is further subdivided using k-fold cross-validation with $k = 5$ for classification tasks.

Dataset	Train		Test	
	Normal	**Anomaly**	**Normal**	**Anomaly**
CSIC2010	36000	0	36000	25065
UNSW-NB15	3514	3998	4978	12394
MALICIOUSURL	278353	0	66488	66448
ISCXURL2016	21085	78134	14293	51854

We briefly summarize the corpora used in our experiments. The train/test split (Table 3.1) follows the authors recommendations whenever available. For two datasets, we deliberately exclude anomalous samples from the training portion (some of which will be later used for finetuning). In such settings, strong BoW performance indicates reliance on token-count differences in normal traffic and on OOV positions. Although fastText and RoBERTa mitigate OOV to some extent, any coverage gaps still surface in the resulting vectors and should be reflected in downstream behavior.

CSIC2010

The CSIC2010 HTTP corpus [1] [109] provides ready-to-use request logs spanning attacks such as SQL injection, buffer overflow, information gathering, file disclosure, CRLF, and XSS, all targeted at a single e-commerce application. The labels are binary. We keep the original split and do not include anomalous samples in the training. The results obtained show that representation learning does not require anomalous traffic to yield competitive classifiers and can still be used in various other tasks.

[1] https://www.tic.itefi.csic.es/dataset/

UNSW-NB15

UNSW-NB15 [2] [110] is a broad intrusion dataset from which we extract the HTTP subset and pair it with the provided labels: *Analysis*, *Backdoor*, *DoS*, *Exploits*, *Fuzzers*, *Generic*, *Normal*, *Reconnaissance*, *Worms*. Many studies rely on author-provided features or custom extraction pipelines; our results are not directly comparable because we restrict attention to HTTP.

MALICIOUSURL

MALICIOUSURL [3] is a binary collection of benign and malicious URLs, popularized with a TFIDF + Logistic Regression baseline and used in [97]. We construct our own split and, mirroring CSIC2010, exclude anomalous samples from training, reflecting realistic scenarios where many attack types are unseen a priori.

ISCXURL2016

ISCXURL2016 [4] [111] aggregates diverse malicious URLs with labels: *Benign*, *Spam*, *Phishing*, *Malware*, *Defacement*. We prepare the split for our experiments.

3.1.2 Basic classification results

Table 3.2: Overall results (mean over five k-folds) for the Random Forest (RF) classifier. The scores are very stable – the standard deviation is always below 1%.

Dataset	BoW			RoBERTa		
	F1	FPR90	FPR99	F1	FPR90	FPR99
CSIC2010	0.99	0.00	0.01	0.98	0.00	0.03
UNSW-NB15	0.98	0.01	0.08	0.99	0.01	0.05
MALICIOUSURL	0.94	0.03	0.30	0.95	0.02	0.17
ISCXURL2016	1.00	0.00	0.00	1.00	0.00	0.00

Table 3.2 shows the classification results. The key observation is that the simplest method, BoW, consistently performs on par with or even better than RoBERTa. Despite relying solely on token frequencies and an OOV bucket, it is sufficient to separate benign from malicious traffic. This highlights both the relative simplicity of the benchmark

[2] https://research.unsw.edu.au/projects/unsw-nb15-dataset
[3] https://github.com/faizann24/Using-machine-learning-to-detect-malicious-URLs
[4] https://www.unb.ca/cic/datasets/url-2016.html

datasets and the informativeness of the tokens produced by the tokenizer. Even when vectorizers are trained only on normal samples, they still achieve high scores. The overall performance closely mirrors the results already reported in related studies [97], [101], [102]. Therefore, one could argue that complex models are unnecessary. However, we do not stop here: to truly assess the trustworthiness of learned representations, we must go beyond accuracy and examine *why* these results hold or not.

3.2 Explainability as Audit in the Training Pipeline

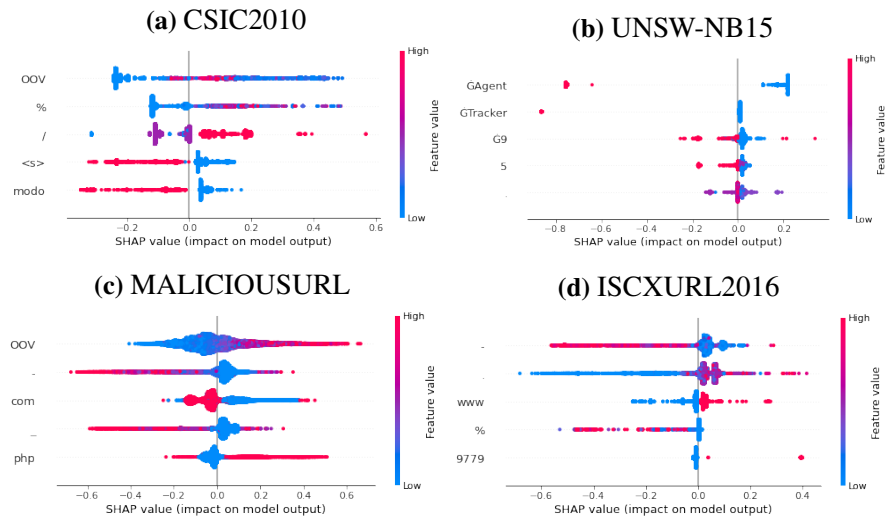


Figure 3.2: Top 5 features for BoW models and their impact on the model's output. The figures were obtained using the SHAP library. The character Ġ is a special token added automatically by the tokenizer and denotes a leading space. SHAP is a local explainability method; the presented global explanation is the average across 200 randomly sampled (stratified) local instances.

Let us begin with the simplest model, BoW. It is the easiest to interpret because every position in the representation directly corresponds to a token frequency rather than to a latent embedding. Given this construction, BoW is also the model most likely to rely on spurious correlations since any token distribution irregularity can be directly exploited by the classifier.

Figure 3.2 illustrates this effect: it presents the five most important tokens according to SHAP values. The vertical axis lists the tokens (e.g. in CSIC2010 these include „OOV", „%", „/", „<s>", and „modo"), while color encodes their frequency in a sample.

Table 3.3: Classification results (5-fold cross-validation) using a Decision Tree trained on the top five BoW features shown in Figure 3.2. Only these five features were used in the classification process.

Dataset	F1	FPR90	FPR99
CSIC2010	0.92	0.03	0.77
UNSW-NB15	0.96	0.01	0.21
MALICIOUSURL	0.78	0.40	0.74
ISCXURL2016	0.96	0.10	0.41

Positive SHAP values indicate correlation with the anomalous class. Some of these tokens can indeed be meaningfully linked to attacks, for instance, the „%" symbol corresponds to percent encoding in URLs. Others, however, appear to be spurious, reflecting quirks of the dataset rather than true attack semantics.

To test whether these few features are sufficient, we built a simple Decision Tree restricted to them. Table 3.3 summarizes the results: most datasets can be classified with very high accuracy using only those features, highlighting their triviality. The exception is MALICIOUSURL, where this shortcut fails. This further underscores both the limitations of BoW and the risk of overreliance on spurious correlations. It is clear that this model is by no means robust and is unlikely to be useful in real-world scenarios.

Remark 1

XAI audits rapidly surface spurious correlations and dataset artifacts.

Next, we investigate how RoBERTa makes its decisions. The process of estimating the importance of every token for transformers is more expensive, so we limited the analysis to a subset of each dataset and present the results for CSIC2010; the observations generalize to the other corpora as well.

We started by selecting a random anomalous request from CSIC2010 and generating its nearest neighbors in the RoBERTa embedding space (Euclidean distance), shown in Table 3.4. The retrieved samples are indeed similar and correspond to injection-style attacks – this is something that is not possible with the BoW and original method, that the space itself can be well integrated (something we will look at in a later section). The SHAP values were then calculated for these examples and Table 3.5 lists the normalized contributions of the attack-related tokens. Unlike BoW, the explanations produced for RoBERTa are much more intuitive: tokens such as "DROP" or percentage-encoded

characters stand out as clear indicators of injection attempts. At the same time, RoBERTa is not free of features that appear irrelevant to the task, such as the token “HTTP”. By contrast, BoW explanations mostly reiterate earlier findings (e.g. high counts of “/” or “Accept-Language: en”), which are dataset-specific rather than semantic cues.

SHAP can also highlight influential parts of a sequence, as illustrated in Figure 3.3. In the case of RoBERTa, the anomalous segments of the request are marked in red, while benign parts are shaded in blue, largely in accordance with the ground truth. This demonstrates that even if RoBERTa does not deliver the highest raw accuracy, its decisions are grounded in interpretable, attack-related features rather than trivial artifacts of the dataset.

Table 3.4: CSIC2010 - 200 nearest neighbours to the “0” sample generated using RoBERTa embeddings and Euclidean distance.

k-th NN	Selected Samples
0	/entrar.jsp?errorMsg=Credenciales+incorrectas%27%3B+DROP+TABLE+usuarios%3B+SELECT+*+FROM+datos+WHERE+nombre+LIKE+%27%25
1	entrar.jsp?errorMsg=Credenciales+incorrectas%27%3B+DROP+TABLE+usuarios%3B+SELECT+*+FROM+datos+WHERE+nombre+LIKE+%27%25
5	/entrar.jsp?errorMsg=%27%3B+DROP+TABLE+usuarios%3B+SELECT+*+FROM+datos+WHERE+nombre+LIKE+%27%25
10	/vaciar.jsp?B2=Vaciar+carrito%27%3B+DROP+TABLE+usuarios%3B+SELECT+*+FROM+datos+WHERE+nombre+LIKE+%27%25
15	/caracteristicas.jsp?id=%27%3B+DROP+TABLE+usuarios%3B+SELECT+*+FROM+datos+WHERE+nombre+LIKE+%27%25
25	/pagar.jsp?modo=insertar&precio=%27%3B+DROP+TABLE+usuarios%3B+SELECT+*+FROM+datos+WHERE+nombre+LIKE+%27%25&B1=Confirmar
50	/anadir.jsp?id=2&nombre=Jam%F3n+Ib%E9rico&precio=100&cantidad=%27%3B+DROP+TABLE+usuarios
100	/pagar.jsp?modo=insertar&precio=6987%27%3Bwaitfor+delay+%270%3A0%3A15%27%3B--&B1=Pasar+por+caja
200	/anadir.jsp?id=1&nombre=Vino+Rioja%27INJECTED_PARAM&precio=85&cantidad=7&B1=A%F1adir+al+carrito

Remark 2

Explanations should be presented in a form accessible to operators, such as token-level highlights, so that human experts can easily inspect and validate model decisions.

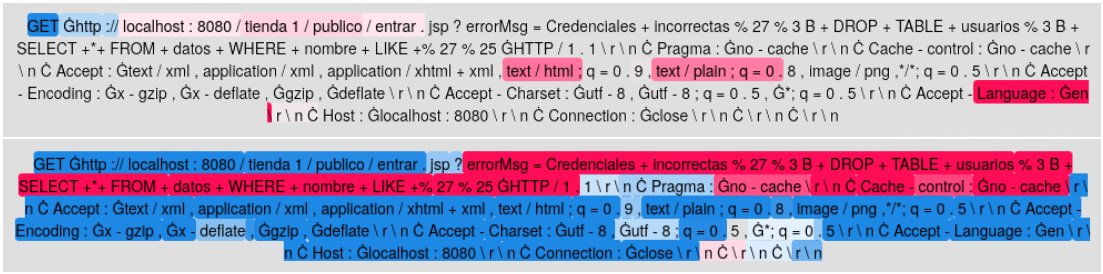


Figure 3.3: Sample "0" with colored token attributions obtained using SHAP. Top: BoW model; bottom: RoBERTa. Red indicates tokens that support the anomalous (positive) label, while blue denotes tokens that support the normal (negative) label.

Table 3.5: Most important tokens related to the positive class (anomalies) for RoBERTa and Bag-of-Words (BoW) models.

CSIC2010 (BoW)			
token	score	token	score
=	0.41	1	0.18
\	0.38	Language	0.18
/	0.38	Ġx	0.18
-	0.32	html	0.16
text	0.31	,	0.13
;	0.28	plain	0.12
.	0.24	deflate	0.09
Ġen	0.23	Ċ	0.08
q	0.20	n	0.08
:	0.20	r	0.08
0	0.19	modo	0.07
&	0.18	tienda	0.07

CSIC2010 (RoBERTa)			
token	score	token	score
%	0.58	TABLE	0.07
+	0.43	usuarios	0.06
3	0.25	precio	0.06
B	0.24	DROP	0.06
&	0.21	A	0.05
27	0.16	22	0.05
cache	0.14	D	0.05
Ġno	0.14	SELECT	0.05
2	0.13	FROM	0.05
nombre	0.08	+*+	0.05
ĠHTTP	0.08	datos	0.05
+%	0.07	WHERE	0.05

3.3 Representation Centric Evaluation

After the audit in Section 3.2, we assess whether the learned spaces themselves are *trustworthy*: do they organize samples in a way that supports unsupervised structure and screening of novel inputs, or do they only deliver headline accuracy? We therefore evaluate (i) clustering quality and (ii) out-of-distribution (OOD) detection. The aim is to turn embeddings into objects that we can *measure* and *improve* in the next section.

3.3.1 Clustering Evaluation

We assess whether the embedding space exhibits an unsupervised class structure by running *kmeans* on RoBERTa embeddings (because they work best from the XAI audit perspective). To avoid coupling scores to the (possibly mismatched) number of ground-truth classes, we sweep the number of clusters $k \in 2, \ldots, 10$ and report the *best* result per dataset. Quality is summarized with *ARI* and *AMI* (as defined earlier).

Table 3.6: K-means on RoBERTa embeddings. Best score over $k \in \{2, \ldots, 10\}$. Metrics are **ARI / AMI** (higher is better).

Dataset	ARI	AMI
CSIC2010	0.118	0.068
ISCXURL2016	0.158	0.171
MALICIOUSURL	0.340	0.385
UNSW-NB15	**0.590**	**0.471**

UNSW-NB15 shows the strongest emergent structure (0.59/0.47), indicating good alignment between classes and geometry. *MALICIOUSURL* exhibits mid-range cohesion (0.34/0.39), often sufficient for analyst triage or weak supervision. *ISCXURL2016* and *CSIC2010* remain entangled (both ARI < 0.18): despite strong supervised F1 elsewhere, their unsupervised geometry is weak – an echo of XAI audits that surfaced shortcut features and artifacts. Such results are even more surprising because selecting the most similar samples (using the same Euclidean distance measure) seemed to work. Such a simple evaluation method provides another essential metric that every model should be able to satisfy.

Remark 3

Low ARI/AMI in the face of high supervised accuracy is an early warning: the model may be exploiting shortcuts rather than learning meaningful geometry, motivating the space-improvement steps that follow.

3.3.2 Out-of-Distribution (OOD) Detection

Generalization asks whether a model keeps high label accuracy on new but *related* data; OOD detection asks whether we can *flag* inputs drawn from a different distribution

altogether. These are complementary: a model may generalize well while still providing useful OOD signals – and vice versa. Trustworthy systems need both. We begin with OOD detection as the next evaluation measure, while leaving a return to the broader question of generalization for later.

Using RoBERTa embeddings, we score test points utilizing two post-hoc detectors: *kNN distance to ID* and a *Mahalanobis score* computed from the ID mean and (regularized) covariance in the embedding space. We evaluated OOD in three regimes – *close* (the same modality, similar semantics), *mid* (related but distinct modality/task) and *far* (random background). Tables 3.7-3.8 summarize the mappings for the two ID roots used below. Experiments on the remaining datasets (UNSW-NB15 and CSIC2010) showed that OOD performs virtually perfectly for each method; therefore, they are not presented as they do not contribute anything to the study.

Table 3.7: OOD regimes for ID=MALICIOUSURL.

OOD set	**Regime**
ISCXURL2016	close (same domain: URLs)
CSIC2010	mid (HTTP requests)
UNSW-NB15	mid (HTTP requests)
far_ood	far (random background)

Table 3.8: OOD regimes for ID=ISCXURL2016.

OOD set	**Regime**
MALICIOUSURL	close (same domain: URLs)
UNSW-NB15	mid (HTTP requests)
CSIC2010	mid (HTTP requests)
far_ood	far (random background)

With *MALICIOUSURL* as the ID dataset, the detectors behave quite differently. kNN performs well on mid-OOD. but its performance drops sharply for close-OOD. Mahalanobis shows the opposite trend: it is weak on close- and mid-OOD but achieves

Table 3.9: OOD detection for MALICIOUSURL as in-distribution (ID). Reported metrics are AUROC and TNR/FPR at 95% TPR. The prefix in each curve indicates the OOD source.

Method	OOD source	AUROC	TNR@95%	FPR@95%
kNN	CSIC2010	0.857	0.000	1.000
kNN	ISCXURL2016	0.639	0.058	0.942
kNN	UNSW-NB15	0.949	0.807	0.193
kNN	far-ood	0.742	0.560	0.440
Mahalanobis	CSIC2010	0.205	0.000	1.000
Mahalanobis	ISCXURL2016	0.503	0.088	0.912
Mahalanobis	UNSW-NB15	0.371	0.041	0.959
Mahalanobis	far-ood	0.968	0.810	0.190

Table 3.10: OOD detection for ISCXURL2016 as in-distribution (ID). Reported metrics are AUROC and TNR/FPR at 95% TPR.

Method	OOD source	AUROC	TNR@95%	FPR@95%
kNN	CSIC2010	0.998	1.000	0.000
kNN	MALICIOUSURL	0.981	0.888	0.112
kNN	UNSW-NB15	0.994	0.984	0.016
kNN	far-ood	0.985	0.920	0.080
Mahalanobis	CSIC2010	0.998	1.000	0.000
Mahalanobis	MALICIOUSURL	0.950	0.765	0.235
Mahalanobis	UNSW-NB15	0.948	0.565	0.435
Mahalanobis	far-ood	0.999	1.000	0.000

the best results on *far-OOD*, confirming that distant distributions are best handled by Mahalanobis. At the same time, FPR@95% rises markedly for close and mid-OOD (e.g. kNN or Mahalanobis against CSIC2010 or ISCX2016), indicating poor rejection when operating at high recall.

For ISCXURL2016 as ID, the picture is more balanced. Both kNN and Mahalanobis yield nearly perfect AUROC and TNR when CSIC2010 is used as OOD and still achieve strong separation on UNSW-NB15. Even MALICIOUSURL – previously the most difficult case-now produces a high AUROC with reasonable TNR. The weakness appears again in FPR@95%, which increases for some OOD sources (e.g. kNN on MALICIOUSURL: 0.1122; Mahalanobis on MALICIOUSURL/UNSW-NB15: 0.235/0.435), in contrast to the near-zero values observed for CSIC2010 or far-OOD.

Overall, these contrasting patterns confirm the need for a *portfolio-based view* of OOD evaluation: performance must be checked with multiple detectors and operating points, rather than relying on a single headline metric.

Remark 4

The choice of OOD score should match the geometry of the embedding space. Local density measures such as kNN are effective when in-distribution data forms several clusters or has an irregular shape. Global scores like Mahalanobis work better when the distribution is closer to a single, roughly elliptical cluster (after whitening) or when OOD data represents a large global shift. In practice, if the ID distribution is clearly multi-modal, it is safer to use class- conditional Mahalanobis or simply rely on kNN.

3.4 Improving the Space

Guided by the audits and representation tests from the previous sections, we now *reshape the embedding space* with small, targeted feedback. The aim is to turn brittle or entangled representations into spaces that better support clustering [112] and OOD detection, without requiring large-scale labeling.

Active learning offers one way to reduce annotation cost by asking humans to label only the most informative items. A central element is the *selector*, the strategy that decides which samples (or pairs) should be presented to an annotator. Prior work has explored this systematically in the context of human-in-the-loop machine learning [113], and our own study [114] showed that while informed selectors (e.g. *closest*, targeting ambiguous boundary cases) often yield gains, even a simple *random* selector can remain surprisingly competitive-particularly when the training objective itself mines hard pairs within a batch. This design choice underlies our setup: we keep a strong *random* baseline and treat more sophisticated selectors as optional refinements.

Table 3.11 illustrates a simple but important lesson: even a handful of annotations can make a difference. With roughly 100 labeled pairs, clustering quality is already improving noticeably, and adding more labels continues to strengthen the results. The benefit is most visible on challenging datasets such as *20News*, where clustering accuracy (AMI) almost doubles compared to the best unsupervised method. In practice, this means that small, carefully placed feedback can reshape representations in a meaningful

way without requiring a large or costly annotation effort.

Remark 5

A small budget of well-chosen pairwise constraints can substantially improve structure with minimal annotation cost; we recommend starting with the *closest* selector and switching only if gains plateau.

Remark 6

Most of the gain arrives early: the first 100-300 pairs deliver the steepest AMI/ARI improvements. Larger budgets refine clusters with diminishing returns.

Table 3.11: Comparison of active learning (closest selector) to an unsupervised baseline using the AMI metric. Values in parentheses indicate relative improvement over the best baseline. Results are from our previous work [114].

#Ann.	AGNews	20News	8tags	Wiki34
100	0.640 (+11%)	0.496 (+84%)	0.450 (+60%)	0.807 (+7.3%)
300	0.677 (+17%)	0.498 (+85%)	0.454 (+62%)	0.823 (+9.5%)
500	0.689 (+20%)	0.525 (+95%)	0.469 (+67%)	0.840 (+11.7%)

We expect a similar mechanism to support our target tasks. However, in our case, we do not limit training to contrastive objectives alone. This joint setup is designed to yield (i) clearer, analyst-facing visualizations, (ii) higher neighborhood purity for clustering, and (iii) more separable boundaries for OOD screening. Motivated by these goals, we implemented a lightweight variant of the simple, selector-friendly, and easy to-replicate framework.

3.4.1 Audit-Guided Representation Refinement

We start from a pretrained RoBERTa encoder and *reshape* its embedding space with a new objective that pulls together semantically similar requests and pushes apart dissimilar ones. Intuitively, this produces more coherent clusters and cleaner decision boundaries while keeping the procedure lightweight and reproducible. For an input sequence x, RoBERTa provides hidden states per-token $\{h_\ell(x,t)\}_{\ell=1}^{L}$. We follow a

simple, robust recipe: concatenate the last four layers and mean-pool over tokens,

$$z(x) \;=\; \mathrm{mean}_t\big[h_{L-3}(x,t)\|h_{L-2}(x,t)\|h_{L-1}(x,t)\|h_L(x,t)\big] \in \mathbb{R}^{4H}.$$

Before computing distances, we ℓ_2-normalize z, which bounds Euclidean distances to $[0, 2]$ and stabilizes training.

We then form training pairs (x_i, x_j, y_{ij}) using labels originating from the same training corpus on which the encoders were previously trained completely unsupervised (no labels were used at that stage). A pair is *positive* ($y_{ij} = 1$) if both elements share the same class and *negative* ($y_{ij} = 0$) otherwise; positives are sampled with probability $p = 0.5$). For variants intentionally pretrained on *benign-only* traffic, the current pairing step additionally draws anomalous examples that the representation model has *never seen* before.

Given normalized embeddings $z_i = z(x_i)$ and $z_j = z(x_j)$ with $d_{ij} = \|z_i - z_j\|_2$ (bounded in $[0, 2]$), we use a symmetric Euclidean contrastive loss with a fixed margin $m = 0.5$:

$$\mathcal{L}_{\text{con}} \;=\; \tfrac{1}{2}\Big(y_{ij}\, d_{ij}^2 \;+\; (1 - y_{ij})\,\big[\max(0,\, 0.5 - d_{ij})\big]^2\Big).$$

To preserve useful structure from the base encoder while reshaping geometry, we add an anchor distillation term that pulls the student toward a frozen teacher (the original RoBERTa). With teacher embeddings t_i, t_j (pooled and normalized identically),

$$\mathcal{L}_{\text{distill}} = \tfrac{1}{2}\big(\|z_i - t_i\|_2^2 + \|z_j - t_j\|_2^2\big).$$

Let θ denote the trainable parameters of the student encoder, and let θ_0 be the same parameters *at initialization* (a snapshot of the pretrained RoBERTa before fine-tuning). We penalize drift from this starting point using L2 Starting Point regularization (L2-SP) [115]:

$$\mathcal{L}_{\text{L2SP}} \;=\; \lambda \sum_k \big\|\theta_k - -\theta_{0,k}\big\|_2^2.$$

In practice, setting λ=0 or freezing lower layers already provides sufficient stability. The total objective is

$$\mathcal{L} \;=\; \alpha_t\, \mathcal{L}_{\text{con}} \;+\; \beta\, \mathcal{L}_{\text{distill}} \;+\; \mathcal{L}_{\text{L2SP}}.$$

When beneficial, the original RoBERTa MLM objective can be retained alongside these terms, but our core results rely on the contrastive and distillation signals.

Evaluation. We evaluate at increasing *label budgets* defined as proportions of the original pretraining corpus: $\rho \in \{0.1, 0.2, \ldots, 0.5\}$. For each ρ, we sample a stratified labeled subset $\mathcal{L}_\rho$, derive contrastive pairs from class labels (positives: same class; negatives: different), and fine-tune the encoder as described above. This protocol *emulates a human-in-the-loop workflow*: as more data are annotated, we observe how representation quality and detection behavior evolve, deciding when to stop, continue, or adjust the selection strategy.

We then re-evaluate the learned space using the previously introduced representation-centric metrics, and, as two small surprises, add *OOD generalization* and *MDS*. The former tests whether boundaries and scores transfer to unseen OOD sources; the latter squeezes the space into 2D to sanity-check that separation reflects geometry rather than plotting artifacts.

3.4.2 New Representation Quality

Classification stability

Contrastive fine-tuning leaves the classifier scores broadly stable (Table 3.12), with changes concentrated where the baseline was weakest. On CSIC2010 the scores slightly degrade (F1 drops by ≈0.02, but FPR@90% increases by ≈0.006–0.008), suggesting that reshaping the space can unlearn dataset-specific shortcuts that benefited the baseline. For ISCXURL2016 the baseline is already near ceiling; fine-tuning yields tiny but consistent F1 gains (up to +0.003) while keeping FPR@90% essentially at zero. The largest benefit appears on MALICIOUSURL: F1 improves monotonically with the fine-tuned fraction (up to +0.025), and FPR@90% drops markedly, indicating cleaner separation of benign vs. malicious URLs. UNSW-NB15 shows modest and steady gains.

Overall, the classification picture is consistent with our goals: we do not optimize for headline accuracy, yet we maintain or improve it where the baseline struggled (MALICIOUSURL), avoid harming already-solved cases (ISCXURL2016), and expose where gains may trade off with reliance on spurious cues (CSIC2010). That said, modest improvements, and the observed behavior of F1 and FPR@90%, are not persuasive on their own. To build a stronger case, we now turn to the representation-centric metrics introduced earlier where the impact of reshaping the space should be more evident.

Table 3.12: Classification after contrastive fine-tuning. We report *F1* and *FPR@90%*; deltas in parentheses are relative to the baseline (RF trained on *frozen* RoBERTa embeddings).

Dataset	(FT fraction)	F1 (Δ vs. base)	FPR@90% (Δ vs. base)
CSIC2010 (baseline: F1=0.980, FPR@90=0.001)			
	10%	0.962 (−0.018)	0.009 (+0.007)
	20%	0.959 (−0.021)	0.009 (+0.008)
	30%	0.960 (−0.020)	0.009 (+0.008)
	40%	0.962 (−0.018)	0.008 (+0.006)
	50%	0.966 (−0.014)	0.005 (+0.004)
ISCXURL2016 (baseline: F1=0.998, FPR@90=0.000)			
	10%	0.998 (+0.002)	0.000 (+0.000)
	20%	0.999 (+0.003)	0.000 (+0.000)
	30%	1.000 (+0.003)	0.000 (+0.000)
	40%	1.000 (+0.003)	0.000 (+0.000)
	50%	1.000 (+0.003)	0.000 (+0.000)
MALICIOUSURL (baseline: F1=0.9544, FPR@90=0.0173)			
	10%	0.957 (+0.002)	0.015 (−0.002)
	20%	0.967 (+0.013)	0.008 (−0.010)
	30%	0.972 (+0.018)	0.005 (−0.012)
	40%	0.977 (+0.022)	0.004 (−0.014)
	50%	0.980 (+0.025)	0.003 (−0.015)
UNSW-NB15 (baseline: F1=0.9810, FPR@90=0.0203)			
	10%	0.980 (−0.001)	0.021 (+0.000)
	20%	0.982 (+0.001)	0.020 (−0.000)
	30%	0.987 (+0.006)	0.020 (−0.000)
	40%	0.988 (+0.007)	0.019 (−0.001)
	50%	0.988 (+0.007)	0.018 (−0.003)

Remark 7

The contrastive + distillation objective does not inherently remove dataset shortcuts. If positive and negative pairs share the same nuisance patterns, the model may keep those cues while merely widening margins. In our setup, random pairing across strata, partial freezing, and teacher anchoring add nuisance diversity and typically reduce shortcut reliance. We explicitly optimize for *robustness*, not peak accuracy: improved OOD separation and stable ROC-derived signals (e.g. FPR@90%) are preferred even if F1 or accuracy move slightly. To strengthen this effect, sample positives that span nuisance attributes and mine hard negatives.

Clustering Evaluation

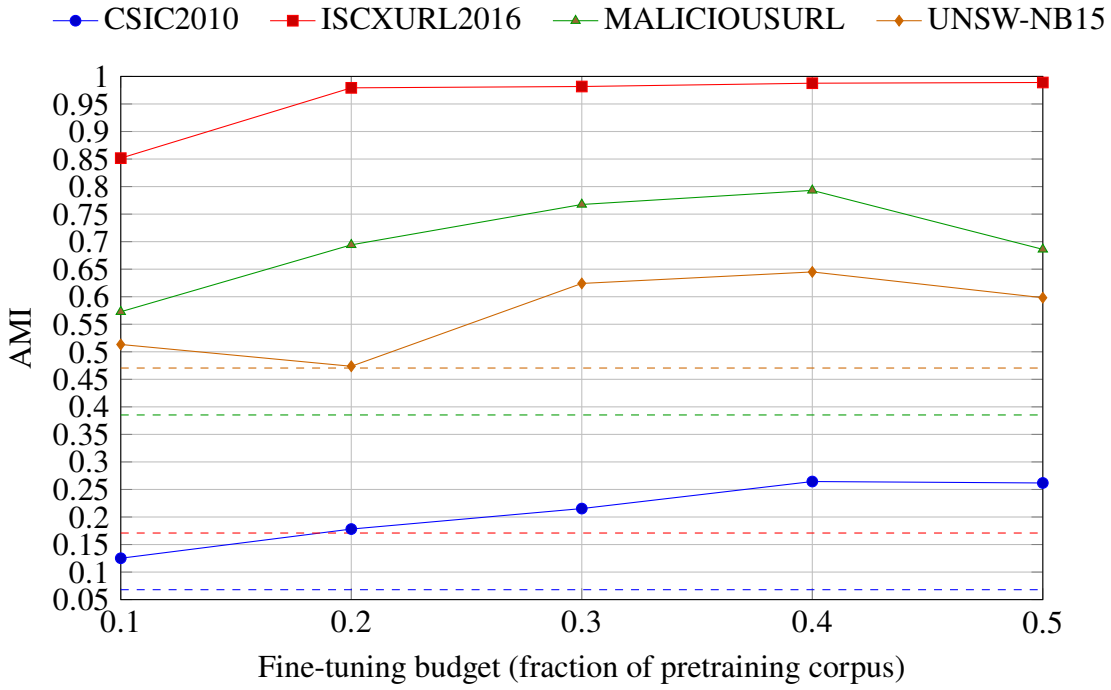


Figure 3.4: Clustering quality (AMI) vs. fine-tuning budget. Dashed lines mark dataset-specific baselines. The taller plot and dense y-ticks highlight incremental gains.

Contrastive fine-tuning substantially improves cluster structure (Table 3.13 and Figure 3.4). On *ISCXURL2016* the gains are dramatic: ARI rises from 0.158 to aproximetly 0.997, and AMI from 0.171 to 0.989, meaning kmeans nearly recovers ground-truth labels. *MALICIOUSURL* shows a steady increase to ρ = 40% (ARI=0.873, AMI=0.793), followed by a drop to 50%, suggesting over-sharpening or local collapse of minority modes. Simmilarly, *UNSW-NB15* improves overall, with the best scores at 40% (ARI 0.749, AMI 0.645) and a slight decrease at 50%. For *CSIC2010* the gains are modest but consistent (from ARI=0.118 and AMI=0.068 to 0.288 and 0.264), which aligns with its heterogeneous attack mix and earlier evidence of spurious cues. Improvements are not strictly monotonic with the label budget. Late stage declines indicate that „more feedback" can over-tighten neighborhoods.

Remark 8

Stop early when AMI/ARI plateaus or starts to fall, and tame over-sharpening by lowering the contrastive margin, reducing the positive-pair ratio, freezing more lower layers, or adding a stronger distillation weight. High AMI with robustness-friendly OOD metrics is the target – not maximal tightening of clusters. This is good news in practice: as additional (often human-annotated) data arrives, there is no need to push to 100% annotations a well-timed partial pass can deliver most of the benefit while saving labeling effort.

Table 3.13: K-means on RoBERTa embeddings before and after contrastive fine-tuning. For each dataset and finetuning budget we report the *best* ARI/AMI over $k \in \{2, \ldots, 10\}$.

Dataset	Budget	ARI	AMI
CSIC2010	Baseline	0.118	0.068
	10%	0.167	0.125
	20%	0.157	0.178
	30%	0.270	0.215
	40%	**0.288**	**0.264**
	50%	0.287	0.262
ISCXURL2016	Baseline	0.158	0.171
	10%	0.926	0.852
	20%	0.993	0.979
	30%	0.994	0.982
	40%	0.996	0.988
	50%	**0.997**	**0.989**
MALICIOUSURL	Baseline	0.340	0.385
	10%	0.681	0.573
	20%	0.791	0.694
	30%	0.854	0.768
	40%	**0.873**	**0.793**
	50%	0.628	0.686
UNSW-NB15	Baseline	0.590	0.471
	10%	0.657	0.513
	20%	0.473	0.474
	30%	0.738	0.624
	40%	**0.749**	**0.645**
	50%	0.650	0.598

OOD detection

As before, we show detailed plots for two representative ID datasets. We focus on *MALICIOUSURL* and *ISCXURL2016*, because the results for *UNSW-NB15* are already near the ceiling. These results pertain to the models that achieved the highest score in the clustering evaluation.

In Figure 3.5 (MALICIOUSURL as ID), fine-tuning consistently raises AUROC across most OOD sources and increases TNR@95%, with the largest gains for *CSIC2010* and *UNSW-NB15*. ISCXURL2016 remains the toughest close/mid-OOD source. Geometry effects are clearly visible: kNN tends to do better on close/mid-OOD, whereas Mahalanobis excels on far-OOD. Figure 3.6 (ISCXURL2016 as ID) shows that the already strong baselines move even closer to the perfect score, while TNR@95% stays near the ceiling in OOD sources. Mahalanobis closes the gap on far-OOD, and kNN retains a slight edge for closer shifts. The accompanying tables (Table 3.14 and 3.15) mimic these plots but provide exact values and deltas versus baseline. Taken together, the figures and tables support a portfolio view: detector choice should match embedding geometry and fine-tuning reshapes the space in ways that improve OOD screening without sacrificing the strong cases.

Table 3.14: OOD after fine-tuning for *MALICIOUSURL* as in-distribution (ID). We show value and delta vs. baseline in parentheses. Reported metrics are AUROC and TNR@95%.

Method	OOD source	AUROC (Δ)	TNR@95 (Δ)	FPR@95 (Δ)
kNN	CSIC2010	0.993 (+0.135)	1.000 (+1.000)	0.000 (−1.000)
kNN	ISCXURL2016	0.813 (+0.174)	0.401 (+0.343)	0.599 (−0.343)
kNN	UNSW-NB15	0.988 (+0.039)	0.987 (+0.180)	0.013 (−0.180)
kNN	far-ood	0.941 (+0.199)	0.755 (+0.195)	0.245 (−0.195)
Mahalanobis	CSIC2010	0.977 (+0.772)	1.000 (+1.000)	0.000 (−1.000)
Mahalanobis	ISCXURL2016	0.830 (+0.327)	0.356 (+0.268)	0.644 (−0.268)
Mahalanobis	UNSW-NB15	0.968 (+0.597)	0.879 (+0.838)	0.121 (−0.838)
Mahalanobis	far-ood	0.992 (+0.025)	0.955 (+0.145)	0.045 (−0.145)

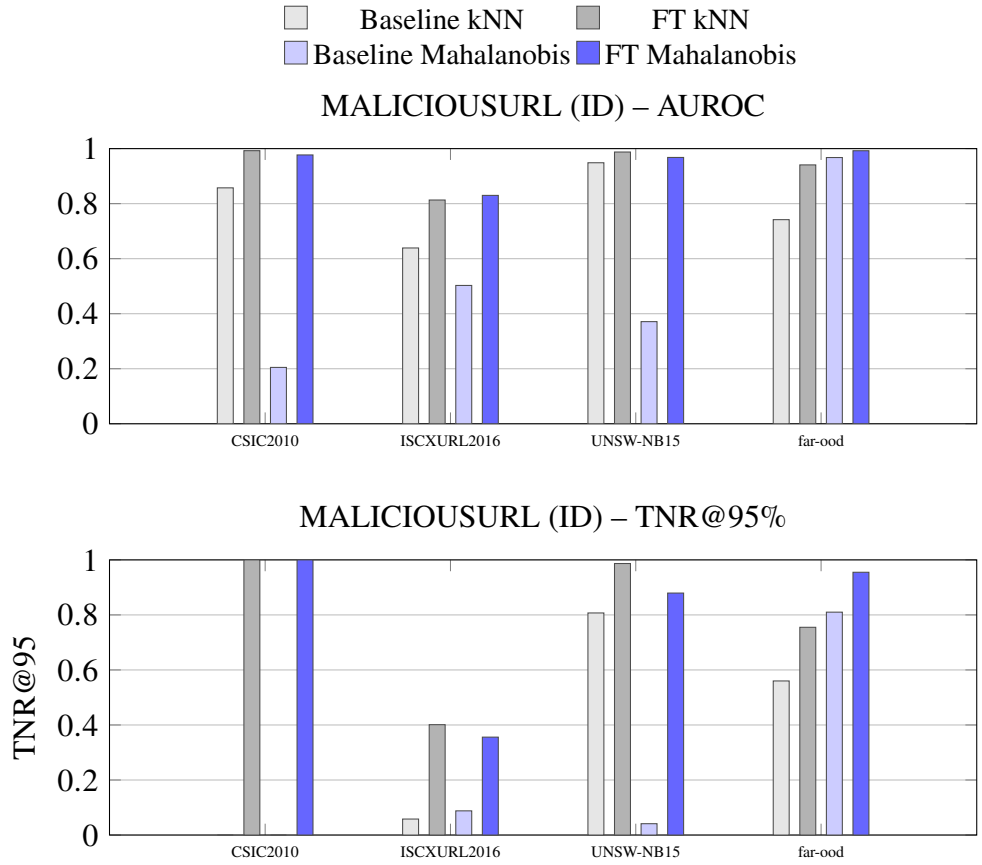


Figure 3.5: *OOD* detection for *MALICIOUSURL* as in-distribution (ID). Bars compare baseline and fine-tuned detectors: kNN (gray) and Mahalanobis (blue). Top: AUROC; bottom: TNR@95%.

Table 3.15: OOD after fine-tuning for *ISCXURL2016* as in-distribution (ID). We show value and delta vs. baseline in parentheses. Reported metrics are AUROC and TNR@95%.

Method	OOD source	AUROC (Δ)	TNR@95 (Δ)	FPR@95 (Δ)
kNN	CSIC2010	0.999 (+0.001)	1.000 (+0.000)	0.000 (+0.000)
kNN	MALICIOUSURL	0.997 (+0.016)	0.999 (+0.111)	0.002 (−0.111)
kNN	UNSW-NB15	0.999 (+0.005)	1.000 (+0.016)	0.000 (−0.016)
kNN	far-ood	0.999 (+0.014)	1.000 (+0.080)	0.000 (−0.080)
Mahalanobis	CSIC2010	0.997 (−0.001)	1.000 (+0.000)	0.000 (+0.000)
Mahalanobis	MALICIOUSURL	0.992 (+0.041)	0.966 (+0.201)	0.035 (−0.201)
Mahalanobis	UNSW-NB15	0.996 (+0.048)	0.994 (+0.429)	0.006 (−0.429)
Mahalanobis	far-ood	1.000 (+0.001)	1.000 (+0.000)	0.000 (+0.000)

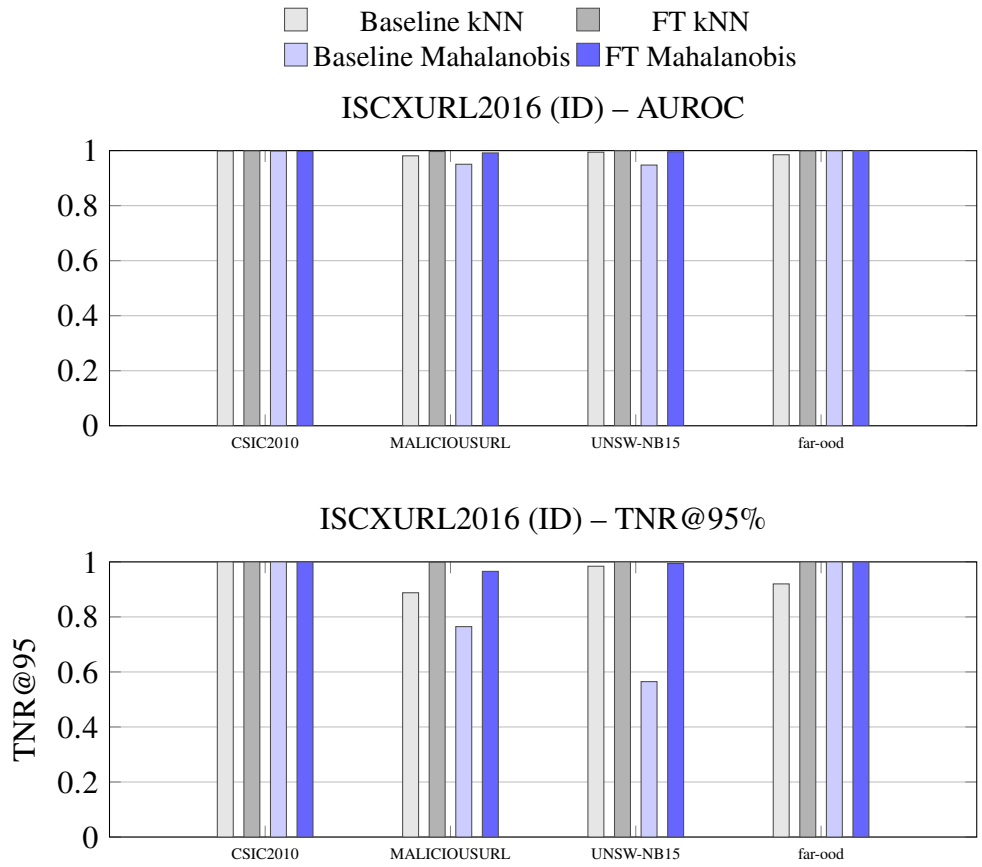


Figure 3.6: *OOD* detection for *ISCXURL2016* as in-distribution (ID). Bars compare baseline and fine-tuned detectors: kNN (gray) and Mahalanobis (blue). Top: AUROC; bottom: TNR@95%.

XAI

Table 3.16 contrasts the most influential tokens for the baseline and the fine-tuned classifier (CSIC2010, RoBERTa embeddings). After fine-tuning, SHAP highlights shift decisively toward attack semantics: SQL keywords (`SELECT`, `WHERE`, `FROM`, `DROP`), operators (=, `LIKE`), and percent-encoding (`%`). Their scores increase by an order of magnitude compared to the baseline, indicating that the decision signal now aligns with meaningful patterns (e.g. injection-like fragments) rather than surface artifacts.

At the same time, not all spurious cues disappear. Tokens such as `cache`/`Cache` still carry non-trivial importance, despite having no intrinsic link to the attacks. This mirrors our earlier caution: reshaping the space improves the *quality* of cues but does not guaranty the absence of dataset-specific shortcuts.

A pragmatic way to suppress residual shortcuts is a *tabu list*: mask or down-weight tokens flagged by XAI as non-semantic (e.g. `cache`) during fine-tuning, then re-audit and iterate. This light-weight loop-audit, mask, refine-tends to preserve high-utility tokens (SQL/encoding hints) while fading dataset quirks.

Remark 9

A pragmatic way to mitigate residual shortcuts is to keep a tabu list: mask or down-weight features that XAI flags as non-semantic (spurious). This generalizes across modalities and can be treated as ordinary preprocessing.

Table 3.16: Comparison of the most important tokens for the baseline and fine-tuned classifier (CSIC2010 dataset, RoBERTa embeddings).

(a) Baseline classifier

Token	Score	Token	Score
%	0.58	TABLE	0.07
+	0.43	usuarios	0.06
3	0.25	precio	0.06
B	0.24	DROP	0.06
&	0.21	A	0.05
27	0.16	22	0.05
cache	0.14	D	0.05
Ġno	0.14	SELECT	0.05
2	0.13	FROM	0.05
nombre	0.08	+*+	0.05
ĠHTTP	0.08	datos	0.05
+%	0.07	WHERE	0.05

(b) Fine-tuned classifier

Token	Score	Token	Score
%	2.811	\	0.303
SELECT	1.334	+	0.278
WHERE	0.976	Ġno	0.271
=	0.684	1	0.240
=+*+	0.649	\n	0.236
FROM	0.519	\r	0.219
=+	0.518	27	0.210
TABLE	0.513	DROP	0.198
Cache	0.372	incorrectas	0.174
cache	0.372	datos	0.164
LIKE	0.349	control	0.161
:	0.307	Ġ:	0.160

OOD generalization

Here we test *cross-dataset transfer* in a strict setting: the RF classifier is trained once on a *source* corpus (using its RoBERTa embeddings) and then kept *frozen*. We apply that same RF to a *target* corpus embedded by the source-trained encoder, either *before* or *after* contrastive fine-tuning. Thus, the vectors and decisions for one data set come from a model learned on another, without retraining of the RF. This isolates whether the reshaped geometry carries across domains.

The trend shown in 3.17 is clear: the generalization of OOD is generally worse-F1 often drops and FPR@90% increases, indicating that fine-tuning tends to establish source-specific cues that do not transfer. There are a few exceptions (e.g. CSIC2010 as target using MALICIOUSURL as source improves FPR@90%), but they are rare.

Table 3.17: Random Forest (RF) kept *frozen* after training on the source. Evaluation is performed on a different target dataset using RoBERTa embeddings. We report *After* and Δ versus *Before*.

Target (test)	Source (train)	F1 (Finetuned, Δ)	FPR@90% (Finetuned, Δ)
CSIC2010	ISCXURL2016	0.582 (+0.000)	0.983 (+0.037)
	MALICIOUSURL	0.582 (+0.000)	0.862 (−0.097)
	UNSWNB15	0.582 (+0.000)	0.727 (+0.002)
ISCXURL2016	CSIC2010	0.795 (−0.048)	0.902 (+0.117)
	MALICIOUSURL	0.758 (+0.004)	0.783 (+0.166)
	UNSWNB15	0.881 (+0.000)	0.992 (+0.023)
MALICIOUSURL	CSIC2010	0.638 (−0.032)	0.885 (+0.080)
	ISCXURL2016	0.669 (−0.018)	0.745 (+0.043)
	UNSWNB15	0.667 (+0.000)	0.915 (−0.004)
UNSWNB15	CSIC2010	0.741 (−0.090)	0.969 (−0.021)
	ISCXURL2016	0.831 (+0.003)	0.987 (+0.118)
	MALICIOUSURL	0.826 (−0.003)	0.930 (−0.056)

In short, within-dataset gains do not automatically translate into OOD generalization with a frozen head (classifier). However, when we *retrain only the classifier* on the target labels using representations from the fine-tuned encoder, F1 and FPR@90% recover to essentially the same levels as in the original within-dataset setting. This indicates that the embedding remains informative across domains; the drop observed with a frozen head is largely a *calibration/boundary misalignment*, not a loss of signal. In practice, this encourages a light head refit (consistent with a human-in-the-loop workflow) that suffices to restore performance while retaining the benefits of the reshaped space for OOD screening and clustering. Importantly, improvements in IID generalization do not necessarily translate into OOD generalization: when applied to external data, the classifier itself requires adaptation, which is often impractical. Further research on ensuring robust OOD generalization is therefore needed, as our fine-tuning clearly does not solve this challenge. However, what is noteworthy is that our analysis makes this

limitation explicit.

Remark 10

Better results on OOD detection do not automatically imply improved OOD generalization, but they at least provide a safeguard: anomalous inputs are less likely to reach the classifier.

Multidimensional Scaling and Similarity

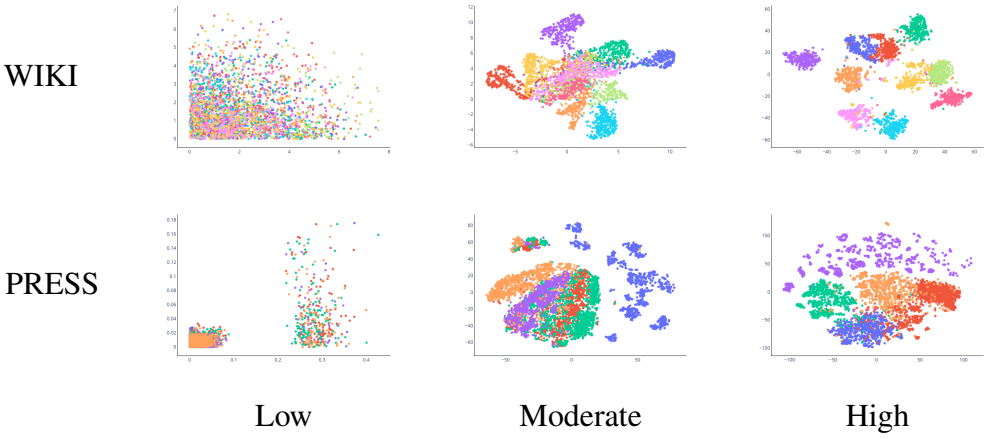


Figure 3.7: Exemplary MDS plots related to the highest, lowest, and median scores for each dataset.

Beyond global clustering metrics, it is useful to inspect how neighborhoods look *locally* in the embedding space. Multidimensional Scaling (MDS) projects high-dimensional points to 2D while preserving pairwise distances as closely as possible, making it a natural fit for analyst-facing dashboards and human-in-the-loop exploration. Following our previous work [116], we treat MDS not only as a visualization tool but also as a quantitative probe: after projecting to 2D, we run kmeans and measure cluster quality with AMI. In this setup, AMI now reflects how well classes separate on the 2D plane, a proxy for how readable and decision-friendly the map is for an analyst.

Concretely, given normalized embeddings $\{z_i\}$ and Euclidean distances $d_{ij} = \|z_i - z_j\|_2$, MDS returns coordinates $\{y_i \in \mathbb{R}^2\}$ that minimize stress, i.e., $(d_{ij} - \|y_i - y_j\|_2)^2$ summed over pairs. We then group $\{y_i\}$ and report AMI (optionally selecting the best

Table 3.18: Clustering in 2D (AMI) *before* and *after* fine-tuning.

Dataset	Method	k (Before)	AMI (Before)	k (After)	AMI (After)
CSIC2010	PCA	8	0.0971	3	0.4588
	t-SNE	2	0.0839	4	0.2808
	UMAP	10	0.1201	6	0.3125
ISCXURL2016	PCA	3	0.2044	3	0.9927
	t-SNE	7	0.2036	6	0.3348
	UMAP	10	0.1623	2	0.7648
MALICIOUSURL	PCA	3	0.3830	2	0.8935
	t-SNE	2	0.4908	2	0.8374
	UMAP	3	0.3156	2	0.8915
UNSW-NB15	PCA	2	0.4549	2	0.5659
	t-SNE	7	0.3329	3	0.4077
	UMAP	4	0.4903	2	0.5837

over a small range of k from 1 to 10). High AMI indicates well-separated „islands" on the map; low AMI corresponds to overlapping blobs-exactly the diagnostic we want for quick, operator-facing checks. In the results that follow, we use this 2D AMI alongside our original-space metrics to confirm that improvements after fine-tuning are visible *and* measurable, and to flag cases where projection distortions or residual entanglement persist. Figure 3.7 provides an idea of how our 2D AMI behaves.

Table 3.18 (with 2D maps for the best $\rho = 40\%$ fine-tuned model) reports AMI *before* and *after* fine-tuning, along with the k chosen by k-means in each case. Post fine-tuning, AMI on the *2D plane* rises sharply for ISCXURL2016 and MALICIOUSURL across projections (e.g. PCA reaches 0.993 and 0.894), indicating that the reshaped space is easier to separate even after dimensionality reduction. CSIC2010 and UNSW-NB15 improve more moderately but consistently, which aligns with their heterogeneous structure and earlier OOD findings. Overall, the *Before* → *After* gains mirror the visuals: low AMI maps to mixed clouds; high AMI to clean islands-useful both diagnostically and for analyst-facing workflows. Figures 3.8, 3.9, 3.10, and 3.11 present the raw 2D visualizations.

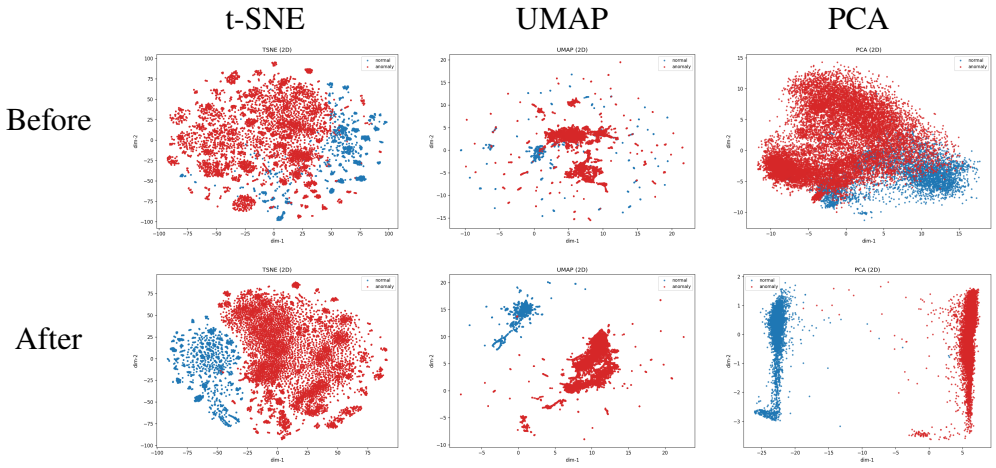


Figure 3.8: ISCXURL2016: Visualization using t-SNE, UMAP, and PCA before and after fine-tuning.

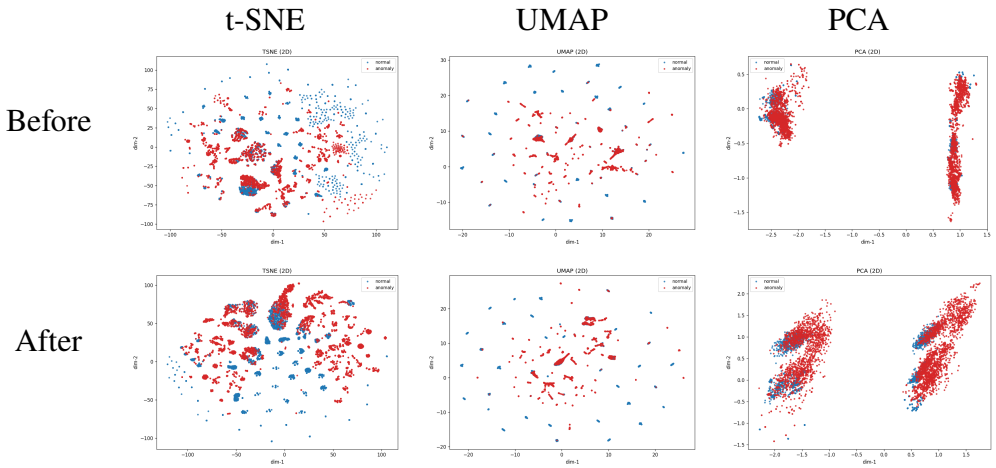


Figure 3.9: CSIC2010: Visualization using t-SNE, UMAP, and PCA before and after fine-tuning.

3.5 Summary

This chapter instantiated the *audit–measure–improve* lifecycle on HTTP/URL classification as a concrete, text-based case study. Starting from a deliberately simple baseline, we showed that bag-of-words models already achieve high accuracy, indicating that the benchmark is relatively easy and that token-level cues suffice for class separation. Explanation-based auditing revealed why: BoW models rely on a small set of irrelevant

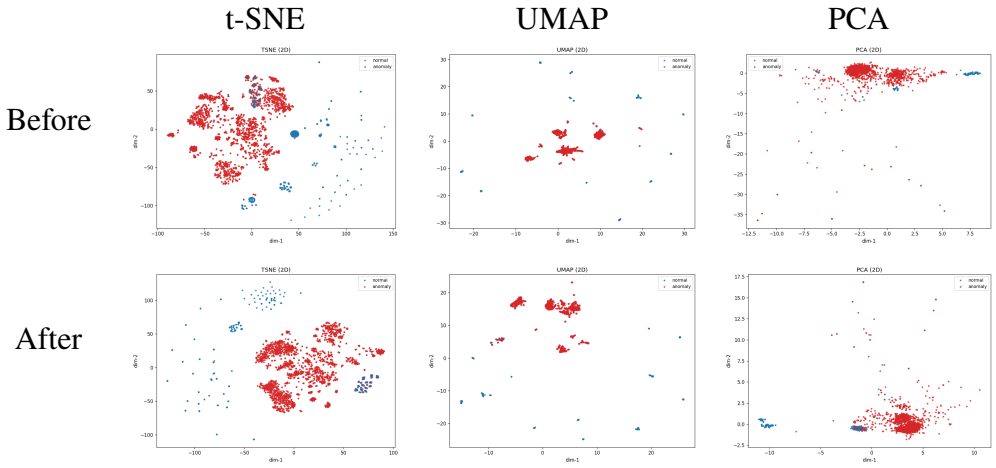


Figure 3.10: UNSW-NB15: Visualization using t-SNE, UMAP, and PCA before and after fine-tuning.

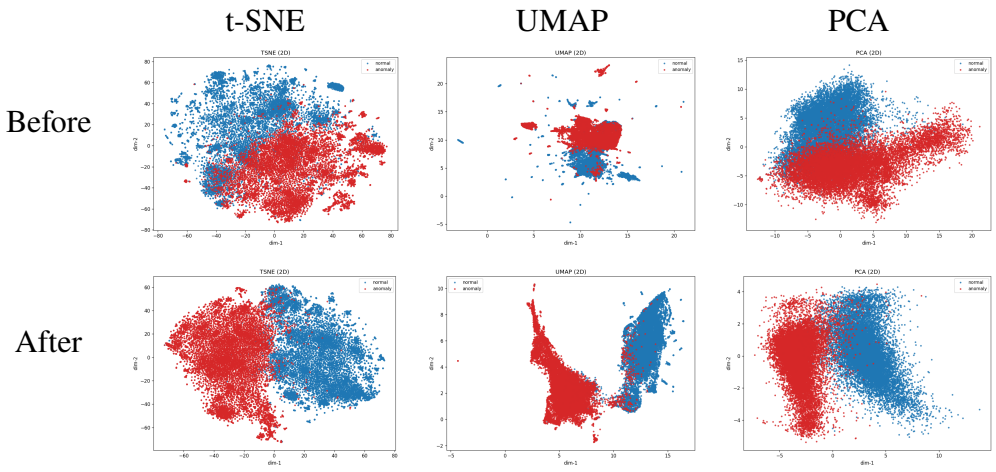


Figure 3.11: MALICIOUSURL: Visualization using t-SNE, UMAP, and PCA before and after fine-tuning.

features, mixing genuinely semantic indicators with spurious shortcuts. In contrast, RoBERTa produced more semantically grounded explanations (e.g., SQL keywords or percent-encoding patterns), while still exhibiting residual shortcut reliance – naturally motivating targeted masking via a compact *tabu list*.

Guided by these audit findings, we introduced a lightweight representation-shaping objective using a limited *human-in-the-loop* labeling budget. The resulting improvements manifested primarily at the geometric level: clustering quality (AMI/ARI) increased,

low-dimensional structure became cleaner and more separable, and OOD detection performance improved – particularly at relevant TNR/FPR operating points. Detector performance aligned with representation geometry: kNN favored locally clustered in-distribution data, whereas Mahalanobis distance was more effective for global, far-OOD shifts.

Cross-dataset *OOD generalization* with a frozen classifier head proved more challenging. Performance drops in F1 and increases in FPR@95% pointed to boundary miscalibration rather than a loss of representational signal. Importantly, retraining only the classifier on a small set of target-domain labels largely restored performance while preserving the improved representation space.

Overall, this study illustrates how trustworthiness emerges from representation quality rather than headline accuracy alone. Within the audit–measure–improve paradigm, modest sacrifices in standard accuracy metrics are justified when they lead to more stable decision boundaries, improved cluster structure, and more reliable OOD behavior.

From a practical standpoint, the improved representation geometry extends the utility of the model. Better structured embeddings support retrieval-style workflows based on representation similarity and enable meaningful global analysis via clustering, as reflected in large post-finetuning gains in ARI and AMI (e.g., on MALICIOUSURL from 0.340/0.385 to 0.873/0.793). At the same time, stronger OOD separation improves the model's ability to flag genuinely novel inputs: on MALICIOUSURL, kNN-based OOD detection reached an AUROC of 0.813 (+0.174) with a TNR@95 of 0.401 (+0.343), reducing reliance on closed-world assumptions and strengthening resilience to previously unseen or zero-day attacks.

Chapter 4

Adversarial Attacks as a Test of Robustness

In recent years, the prevailing narrative has been that explainable AI (XAI) makes classifiers safer by revealing the evidence behind each prediction. Paradoxically, the very explanations that foster trust also expose vulnerabilities. Once an attacker discovers that the *positive* label is highly dependent on a single feature, they can exploit that knowledge to manipulate the output. Empirically, such perturbations require neither gradients nor many attempts, successfully bypassing both rule-based and neural content filters. We are thus living in a dual reality: on one hand, we pursue adversarial robustness; on the other, we deploy explainable AI. Until recently, these goals have been pursued along separate tracks, but emerging research suggests a strong coupling: robustness may promote clearer explanations, while precise explanations can be weaponized.

In [117], the authors were among the first to show a formal connection between adversarial robustness and explainability. They demonstrated that the distance of a sample to the decision boundary limits how much its gradient-based saliency map can deviate from the sample itself. In practice, convolutional neural networks trained with Lipschitz regularization on datasets like MNIST and ImageNet not only became more robust to adversarial attacks, but also produced heatmaps that better matched the actual content of the image. This suggests that interpretability is not just a separate goal, but something that can naturally emerge from robust models.

At the same time, [118] revealed the other side of the coin: the explanations themselves can be fragile, even when the prediction of the model stays the same.

They introduced interpretation-based contradiction examples, crafted not to change the classification outcome but to drastically alter the explanation. In datasets such as CIFAR-10 and ImageNet, small and imperceptible changes were enough to shift the model's attention away from the actual object, undermining saliency, LRP, or DeepLIFT. This highlights a key risk: post-hoc explanations can be silently manipulated without any visible change in model behavior.

Together, these findings reveal a delicate interplay: robustness and explainability are connected, explanations can be manipulated without changing the model's output, and now the output itself can be changed by manipulating what the explanation focuses on. Building on this line of work, [119] propose a CAM-guided attack that first identifies the contextual regions most influential to the decision and then sparsely injects points into those areas, deceiving detectors on KITTI, nuScenes, and Waymo Open, while keeping perturbations visually unobtrusive. This shows that explainability methods, while designed to enhance safety, can also guide precise and transferable attacks.

Remark 11

XAI is dual-use: diagnostic for defenders, actionable for attackers.

We also contribute to this line by studying explanation-guided attacks on text classifiers, both in untargeted and targeted regimes, including black-box pipelines based on large language models [120], [121]. From the perspective adopted in this thesis, adversarial attacks – including the explanation-guided variants proposed here – are not treated solely as security threats, but as diagnostic tools. They form a component of the *audit–measure–improve* loop, enabling systematic stress-testing of models, exposing brittle decision rules, and guiding targeted interventions aimed at improving robustness.

4.1 XAI-guided Untargeted Attacks on Text Classifiers

In this section, we explore how one of the most widely used explainability methods, SHAP scores [67] can be leveraged to guide adversarial attacks. We focus on natural language models due to their broad range of applications, from basic tasks such as token recognition (e.g. named entities or parts of speech) to more complex ones like sentiment analysis and document classification. We demonstrate how learning the importance of individual tokens improves the efficiency and precision of attacks on these tasks. Such attacks further complement our portfolio of methods for stress-testing models beyond

standard accuracy metrics.

The attacks presented here are based on existing methods in adversarial NLP, such as TextFooler [122] and TextBugger [123], and include techniques such as synonym replacement (often with rare or obscure alternatives), insertion or deletion of words modifying intensity, character swaps, misspellings, and symbol substitutions. Although many studies evaluate the effectiveness of such attacks, few attempt to explain or precisely guide them.

Setting. Given a victim text classifier f and an input x, our aim is to craft an adversarial text x^* such that $f(x^*) \neq f(x)$, while preserving the original meaning from a human perspective. Unlike gradient-based approaches, we assume a more restricted setting: the attacker can only observe predicted labels and their associated probabilities.

4.1.1 Algorithm

The core of our approach is a synonym-level adversarial attack, outlined in Algorithm 1. The intuition is straightforward: take an input sentence X with tokens $[w_1, \ldots, w_n]$, exclude named entities to avoid nonsensical replacements, and then generate candidate substitutions for the remaining tokens. Each candidate is applied to X to form a perturbed sentence X', which is validated against two criteria: (i) it must remain semantically close to the original according to a similarity function $Sim(X, X')$ (cosine similarity of the Sentence-BERT embeddings), and (ii) it must change the classifier's prediction from the original label Y. Sentences that pass both checks are collected into the adversarial set X_{adv}.

Substitutions are created using three complementary methods: (i) synonym-based replacement, where w_i is substituted with synonyms drawn from WordNet (English) or plWordNet (Polish), with morphological adjustment via *FindForm* in the Polish case and filtering by cosine similarity above a threshold ϵ_w; (ii) the discard method, where w_i is simply removed, yielding a shorter but often still grammatical sentence; and (iii) the character-discard method, where random characters are deleted from w_i with probability p, introducing lightweight noise. From these options, *CreateCombinations* assembles substitution sets of size s, producing a pool of candidates that balance semantic plausibility and adversarial diversity.

By design, this attack directly mirrors the mechanics of TextFooler but adapts them to inflected languages like Polish, where synonym search is performed on lemmatized

Algorithm 1 Adversarial Attack with WordNet TextFooler

```
Input: Sentence words list X = [w_1, w_2, ..., w_n]
Y sentence class label,
similarity function Sim(.),
sentence similarity bound ϵ,
character deletion probability p,
word similarity bound ϵ_w,
s number of words to substitute,
named entity recognition function NamedEntities(.)
Output: Adersarial examples with changed classifier response X_adv
Initialization: dictionary of words synonyms Synonyms ← {},
set of adversarial examples X_adv ← {}

X ← exclude NamedEntities(X) from X
for each word w_i in X do:
    if discard method then
        Synonyms ← ""
    end if
    if char_discard method then
       Synonyms ← RemoveRandomCharacters(w_i,p)
    end if
    if synonyms method then
        if language polish then
            LemmaSynonyms ← FindLemmaSynonyms(w_i)
            Synonyms ← FindForm(LemmaSynonyms_i)
        end if
        if language english then
            Synonyms ← FindLemmaSynonyms(w_i)
        end if
        Synonyms ← FilterSynonyms(Synonyms, ϵ_w)
    end if
end for

SynonymsCombinations ← CreateCombinations(Synonyms, s)
for each synonyms Synonyms_i in SynonymsCombinations do:
    X' ← X
    for each s_j in Synonyms_i
        X' ← Replace w_j with s_j in X'
    if Sim(X', X) > ϵ and F(X') ≠ Y then
        Add X' to set X_adv
    end if
end for
return X_adv
```

forms and then inflected back to match the grammar of the original token. This significantly reduces the search space and avoids ungrammatical substitutions while

keeping the adversarial signal effective.

Building on the baseline WordNet TextFooler, we introduce an explainability-driven variant where token selection is no longer random or exhaustive but guided by global SHAP importance scores. The idea is to decouple the attack loop from sentence-specific heuristics and instead use precomputed importance rankings that reflect how strongly each token contributes to the classifier's decision. This modification makes the attack both faster (importance values are calculated once on a held-out split) and more precise (it directly targets semantically influential words).

The procedure differs from Algorithm 1 only at the selection stage: instead of iterating over all $w_i \in X$, we pick the top-k tokens from the global importance ranking and align them with their positions in the current sentence. These are then substituted using the same three strategies as before (synonym replacement, deletion, or character-drop). Formally, the XAI-enhanced attack proceeds in three steps: (i) generate global SHAP values with *XaiImportance* and sort them with *SortByImportance*; (ii) select the top tokens k, where k is a hyperparameter controlling the aggressiveness of the attack; (iii) map these tokens back to their indices in X and apply standard substitution routines.

In practice, this extension increases the efficiency of the attack by focusing perturbations on the most impactful parts of the input while retaining generalization: the ranking is model-driven but agnostic to any specific sentence being attacked.

4.1.2 Experiments

To assess the effectiveness of the proposed approach, we conducted experiments on five publicly available datasets. These included two benchmarks focused on sentiment analysis (IMDB and Multi_Emo), two for topic classification (AGNews and Wiki_PL), and one dataset designed for spam detection (Enron_Spam). Three of the datasets were in English, while the remaining two were in Polish.

Our focus here is on these new datasets. Although adversarial attacks could also be applied to the datasets used in previous experiments (e.g. CSIC2010), the setup would need to be considerably more constrained. In such a case, we would be interested primarily in manipulations that flip malicious samples into being misclassified as benign while preserving the underlying harmful intent of the sample itself. This requirement drastically reduces the feasible search space, since many naive synonym substitutions would inadvertently break the semantics of the original attack. As a result, the practical analysis of such scenarios would need to be performed in a more

human-in-the-loop manner, reflecting how an adversary might adapt their attack while retaining its effectiveness.

For all experiments, we used state-of-the-art BERT-based classifiers tailored to the language of each dataset. Each model consisted of a pre-trained BERT encoder followed by a fully connected classification layer. The entire model, both encoder and classifier, was fine-tuned end-to-end. For English datasets, we used the uncased BERT base model [37], while for Polish datasets, we utilized HerBERT [124], a transformer specifically trained on large-scale Polish corpora.

Training was performed on 90% of each dataset's original training set, with the remaining 10% used for model selection. The original test set was further divided into two parts: 90% was reserved for the final evaluation, and 10% was used exclusively to calculate SHAP values. Adversarial examples were generated and evaluated on the final test subset.

4.1.3 Results

Table 4.1: Dataset characteristics: language, number of classes, split sizes, average text length (in words), and BERT accuracy on the test set.

Dataset	Lang	#Classes	Size			Avg. len	ACC [%]
			Train	Test	XAI		
AG_News	EN	4	120 000	6 840	760	38	94.72
IMDB	EN	2	8 000	1 800	200	211	95.17
Enron_Spam	EN	2	31 716	1 800	200	201	98.44
Wiki_PL	PL	34	6 885	2 657	295	186	94.88
Mulit_Emo	PL	2	4 319	972	108	129	98.25

Table 4.1 summarizes the datasets used and the classification performance of the BERT-based models. Although these models are capable of generating contextual embeddings for previously unseen words, the absence of specific terms, such as rare synonyms from WordNet or even common misspellings, remains a key vulnerability that can be exploited in adversarial attacks. This lexical gap contributes to the relative ease with which such models can be fooled.

Table 4.2: Adversarial attack results across five datasets. **Textfooler** and **Textbugger** serve as baselines. Our approach includes three variants: **XAI** (replacing relevant words with synonyms), **XAI-ChD** (removing random characters from relevant words), and **XAI-Discard** (removing relevant words). *Time* denotes processing time for the full test set (in minutes). *Success [%]* is the percentage of successful attacks (similarity ≥ 0.95), and *Success NER [%]* measures successes that altered named entities (our methods preserve NEs).

Attack type	Dataset	Time [m]	Success [%]	Success NER [%]
Textfooler	AG_News	660	16.68	8.60
Textbugger	AG_News	687	**17.46**	10.40
WordNet-XAI	AG_News	128	2.27	0
WordNet-XAI-ChD	AG_News	13	4.25	0
WordNet-XAI-Discard	AG_News	12	2.99	0
Textfooler	Wiki_PL	106	0.34	0.23
Textbugger	Wiki_PL	256	2.56	1.54
WordNet-XAI	Wiki_PL	62	2.94	0
WordNet-XAI-ChD	Wiki_PL	20	9.45	0
WordNet-XAI-Discard	Wiki_PL	17	**10.80**	0
Textfooler	Enron_Spam	4	0.72	0.22
Textbugger	Enron_Spam	3	0.89	0.38
WordNet-XAI	Enron_Spam	36	0.72	0
WordNet-XAI-ChD	Enron_Spam	8	**1.50**	0
WordNet-XAI-Discard	Enron_Spam	7	1.33	0
Textfooler	IMDB	7	4.33	0.11
Textbugger	IMDB	6	4.39	0.55
WordNet-XAI	IMDB	35	4.28	0
WordNet-XAI-ChD	IMDB	12	11.89	0
WordNet-XAI-Discard	IMDB	13	**12.22**	0
Textfooler	Multi_Emo	2	1.34	0
Textbugger	Multi_Emo	2	1.75	0
WordNet-XAI	Multi_Emo	20	1.13	0
WordNet-XAI-ChD	Multi_Emo	5	**5.25**	0
WordNet-XAI-Discard	Multi_Emo	4	4.02	0

The primary objective of the proposed attacks is to degrade the classification accuracy (*ACC*) as much as possible while ensuring that the modified samples remain semantically and syntactically close to the originals. All attacks were conducted on the

test subsets of the datasets, and the corresponding results are provided in Table 4.2.

Table 4.3: Top 5 most significant named entities, parts-of-speech tags, and tokens in the **AG_News** dataset ("XAI" subset only) for each class. Importance is measured as average SHAP value.

Class	NER	Importance	POS tag	Importance	Token	Importance
Sci_Tech	LOC	0.02	CCONJ	0.02	movie	0.55
	PRODUCT	0.01	SPACE	0.02	foxes	0.54
	MONEY	0.00	ADP	0.01	telecommunications	0.52
	QUANTITY	-0.01	PUNCT	0.01	spaceship	0.51
	PERCENT	-0.01	AUX	0.00	photograph	0.47
World	NORP	0.16	PROPN	0.02	nairobi	1.80
	GPE	0.11	ADJ	0.01	nepal	1.73
	LOC	0.05	PUNCT	0.01	jakarta	1.55
	LAW	0.00	PART	0.00	thailand	1.22
	QUANTITY	0.00	X	0.00	jordan	1.21
Business	ORG	0.05	NOUN	0.05	industries	2.47
	FAC	0.05	VERB	0.02	corporate	2.17
	MONEY	0.03	PROPN	0.02	investors	1.84
	LAW	0.02	SYM	0.02	investments	1.83
	PRODUCT	0.02	ADJ	0.01	martha	1.50
Sports	EVENT	0.11	INTJ	0.01	baseball	1.81
	FAC	0.04	DET	0.01	football	1.80
	ORDINAL	0.04	CCONJ	0.01	hockey	1.74
	WORK_OF_ART	0.03	NUM	0.01	tennis	1.61
	PERSON	0.02	PRON	0.01	basketball	1.47

Remark 12

XAI-guided attacks generalize well. Global importance values, computed once, can guide attacks across many inputs, without direct access to specific test samples.

Table 4.3 presents the global importance scores, averaged over samples from the XAI subsets, for the top five named entities (NER), parts of speech (POS tags), and tokens, as determined using the SHAP method. These values are later used to guide the selection of text segments targeted by adversarial modifications. A positive SHAP score indicates that the feature supports the prediction of the studied class, while a negative score suggests that its presence pushes the prediction away from that class.

Among the five datasets used in our experiments, named entities were notably significant only in the case of *AG_News*. In contrast, part-of-speech categories proved relevant primarily for the Polish corpora, in particular, adjectives in the sentiment-oriented *Multi_Emo* dataset. Across all benchmarks, the most influential tokens were often semantically aligned with their associated class. For instance, in *IMDB*, the most positively correlated words for the *positive* class included "outstanding", "heartbreak", and "excellent", while the *negative* class was marked by terms such as "hoax" and "frustrating".

Table 4.2 presents the results of our adversarial attack experiments. For each sample that was originally correctly classified by the model, we attempted to generate adversarial modifications. An attack was considered successful if at least one altered version of the input led to a misclassification.

Although this evaluation strategy is effective for benchmarking, it does not reflect real-world constraints, and many generated modifications may diverge significantly from the original input. To address this, we report only those adversarial samples that maintain high semantic similarity with the original input. Specifically, the column "Success" shows the percentage of successful attacks where the cosine similarity exceeded a 95% threshold. The results demonstrate that our method identifies more adversarial examples than existing tools for most datasets, and does so in less time. The exception is the *AG_News* dataset, where existing methods perform better.

There are two notable differences between our WordNet-XAI-based attack and the TextFooler/TextBugger approaches. First, the latter methods often modify named entities, which are typically excluded in our setup. Second, they tend to alter a larger number of tokens per sample. The column "Success NER" in Table 4.2 quantifies the proportion of successful attacks that involved modifying named entities. This sheds light on the performance gap between approaches and is consistent with our earlier findings from Table 4.3, where we showed that named entities play a significant role in classifying *AG_News* examples.

To better understand the underlying factors that affect performance, we conducted a series of ablation studies, reported in Table 4.4. In row B, we show the results obtained using the WordNet TextFooler method without XAI-based ranking. Although this variant outperforms the XAI-guided versions (rows marked A), it still underperforms

Table 4.4: Ablation study on the **AG_News** dataset. (A) Baseline results from Table 4.2 (processing time and success rate at cosine similarity ≥ 0.95). (B) WordNet Textfooler variant (WordNet-XAI without XAI). (C) Modified methods allowing named entity (NER) substitutions. (D) Doubling the substitution budget.

Group	Attack type	Subst.	Time [m]	Success [%]
A	Textfooler	–	330	16.68
	Textbugger	–	291	**17.46**
	WordNet-XAI	12	128	2.27
	WordNet-XAI-ChD	12	13	4.25
	WordNet-XAI-Discard	12	12	2.99
B	WordNet Textfooler	12	333	9.13
C	WordNet-XAI-ChD NER	12	12	3.89
	WordNet-XAI-Discard NER	12	11	2.63
D	WordNet-XAI-ChD	24	29	4.54
	WordNet-XAI-Discard	24	24	3.07

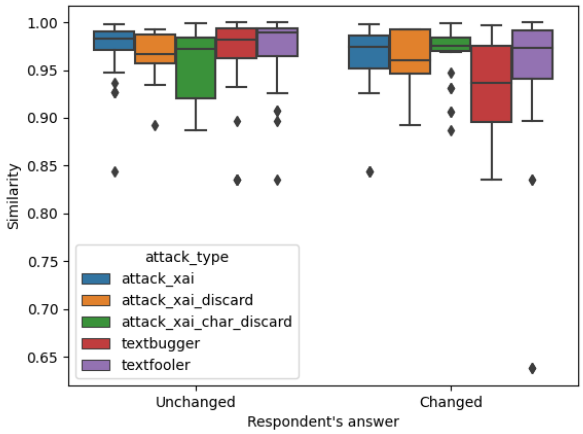


Figure 4.1: Correlation between automated similarity score and human judgments for every evaluated method. Each point corresponds to an evaluated method; the plot quantifies the alignment between the computed similarity and perceived semantic preservation. The data were collected via a survey on the AG_News collection.

the original TextFooler. Its runtime is also considerably higher than the XAI-based methods and comparable to TextFooler itself.

These results suggest two key observations: (1) named entities are particularly critical in *AG_News*, and (2) the importance of global features misgeneralizes their influence. This can be attributed to the prevalence of named entities in the dataset. Our method could benefit from using local (per-sample) explanations instead of a globally precomputed ranking. The increased runtime for non-XAI-based methods stems from the greater number of substitution candidates, whereas XAI-guided attacks focus only on tokens deemed important.

We also experimented with the possibility of enabling named entity modification in two of our variants *WordNet-XAI-ChD* and *WordNet-XAI-Discard*. However, this adjustment did not yield performance improvements, primarily because most named entities do not exist in WordNet [125], making substitution infeasible. Finally, increasing the number of possible substitutions per token offered marginal gains in effectiveness.

4.1.4 Human evaluation

Remark 13

Human judgments align with similarity metrics, mostly: Cosine similarity between sentence embeddings correlates with user perception of meaning preservation, except in the case of character-deletion noise. A more specialized model, such as a fine-tuned Sentence-BERT or a siamese network trained specifically on noisy or adversarial data, would likely provide better alignment with human intuition.

To further investigate the impact of our adversarial modifications, we conducted a user study based on samples from the *AG_News* dataset. Participants were shown two versions of a news article excerpt, one original and one adversarially modified, and were asked to assess whether the meaning of the sample was preserved (*positive*), altered (*negative*), or if they were unsure (*neutral*). A total of 70 respondents took part in the survey, providing 723 valid judgments. Each modified sample was evaluated at least three times.

Table 4.5 summarizes the survey results. The method of highest rank in terms of perceived semantic similarity was *TextBugger*, followed by *WordNet-XAI-ChD*. This highlights the effectiveness of transformation strategies such as character swaps, insertion of visually similar characters, or splitting words, techniques not yet implemented in our approach. A combination of these perturbation types with more structured XAI-guided

Table 4.5: Survey results on the **AG_News** dataset. Seventy participants compared 723 pairs of original and adversarial texts. The *Changed* column reports the percentage of cases where the modified text was judged to belong to a different class.

Attack type	Changed [%]
Textfooler	43.28
Textbugger	23.65
WordNet-XAI	47.17
WordNet-XAI-ChD	37.97
WordNet-XAI-Discard	**54.24**

strategies would likely yield even stronger results, both in terms of human perception and computational efficiency.

The survey also reinforces a second insight, previously hinted at in Table 4.2: replacing named entities with synonyms does not consistently produce natural-sounding modifications. Many incorrectly perceived samples involved changes to proper nouns, such as locations or subjects, suggesting a need for more refined synonym dictionaries tailored to named entities.

Finally, our study revealed a meaningful correlation between cosine similarity scores and human judgment. As shown in Figure 4.1, the lower the similarity score, the more likely the users were to perceive a change in meaning. The main exception was the *CharDiscard* method, where even high-similarity modifications were frequently flagged by respondents. This suggests that the sentence embedding model used for measuring similarity may require further fine-tuning on noisy or perturbed data to better reflect human intuition in this context.

4.2 Targeted attacks with restricted knowledge

Building on the untargeted attacks discussed in the previous section, we next examine the more restrictive case of targeted attacks, where the adversary must coerce the classifier into a specific wrong label. This is described in our work [121].

The new approach is most closely related to the previously described; however, we adapt token-ranking strategy to the targeted setting. The objective is not only to trigger any misclassification, but to systematically redirect input from class A to class B. This tighter constraint reduces the set of viable perturbations and demands a customized

search procedure. Accordingly, we embed the attack within state-of-the-art NLP pipelines, including generative large language models (LLMs), to assess its effectiveness under realistic, high-capacity systems.

4.2.1 Approach

Prompt:
Classify the sentence as 0 (cat) or 1 (dog). Return only the digit:
`<Text>`

Text:
A dog is barking, a cat is meowing, a cow goes "moo".

Result:
Class cat: A dog is barking, a cat is moewing, a cow does muu
Class dog: A dog is barking, a cat is moewing, a cow does muu

Figure 4.2: Example classification prompt and the resulting token-level SHAP explanations for GPT-4o-mini. Red tokens contribute positively to a class, while blue tokens contribute negatively.

The aim of our method is to mount *targeted* attacks: we want the model to produce a specific, pre-chosen *incorrect* label. We achieve this by guiding the perturbation process with class-aware token rankings derived from SHAP scores. By steering the attack along these importance lists, we can systematically push the prediction toward the desired class while keeping the edit budget small.

Differential ranking. Let $R_A = \{(w, S_A(w))\}$ and $R_B = \{(w, S_B(w))\}$ denote the sets of tokens paired with their SHAP scores for classes A and B, respectively. For each token w we compute a *differential score* $\Delta S(w) = S_A(w) - S_B(w)$. Sorting tokens in descending order of $\Delta S(w)$ yields

$$R_{A \to B} = \big\{(w, \Delta S(w)) \mid w \in R_A \cup R_B\big\}.$$

Tokens with large positive $\Delta S(w)$ strongly support class A while contributing little (or even negatively) to class B; they are therefore prime candidates for modification. If $S_B(w) > S_A(w)$, the token already favours class B and is left untouched. By iteratively

editing the top-k items of $R_{A\to B}$ we dampen evidence for A and nudge the model toward B.

Computing Token Importance for LLMs

Large language models accessed through an API expose only limited internals, so we adapt SHAP to this black-box scenario. Given a fixed prompt (see Figure 4.2) and an input text, we create perturbed versions of the text while leaving the prompt unchanged. Each variant is sent to the LLM via the OpenAI API, requesting the generated token sequence together with `logprobs`. Converting log-probabilities to probabilities lets us observe how omitting or altering individual tokens shifts the likelihood of each class. In binary tasks we track the two class tokens, making the SHAP computation efficient. Returned tokens that do not match any class label are ignored during attribution. The resulting per-token SHAP scores feed directly into the differential ranking described above, enabling targeted attacks against black-box LLMs.

4.2.2 Experiments

Our evaluation starts with a BERT classifier fine-tuned for standard text classification. We then benchmark its robustness against zero-shot LLM baseline: GPT-4o-mini, using the prompt format shown in Figure 4.3. As the figure illustrates, even small edits to an input can mislead all three models, yet the generation constraints keep the altered sentences almost indistinguishable from the originals.

4.2.3 Results

Table 4.6: Classification accuracy (ACC) for Wiki_PL and AG_News datasets. All datasets are balanced across classes to ensure fair evaluation. The BERT model is fine-tuned on the respective training sets, while GPT-4o-mini perform zero-shot classification using prompts with natural-language descriptions of each class.

Dataset	BERT ACC [%]	GPT-4o-mini ACC [%]
Wiki_PL	99.00	99.00
AG_News	95.00	85.00

Prompt: Classify sentence into one of the following classes:
0: Articles related to international events, global news, and world affairs. This category includes stories on political events, international conflicts, diplomacy, and relations between countries.
1: This category includes news related to sports events, athletes, match outcomes, developments across various sports, as well as updates on teams and sporting events.
2: Articles in this category cover financial, economic, and market-related news. It includes content on companies, market trends, investments, financial matters, and economic topics.
3: Articles focused on science and technology news. Topics include new technologies, scientific research, discoveries, and trends or innovations in fields such as medicine, IT, computers, and beyond.
Return only a single digit related to class:
<Text>

Text: Panel Urges N.Y. to Pay $14 Billion More for City Schools Court appointed referees recommended the state → commonwealth pay an additional $14 billion over four years to improve New York City schools.

Result: 0 → 2

Figure 4.3: Example of successful directed attacks for AG_News classification (test set) using the GPT-4o-mini model, achieved by altering just a single word in the sentence. The green text indicates the original form, while the red text shows the change introduced by the attack. The importance score of the word "state" is 0.021 for class "0" and −0.019 for class "2". The difference between these values (0.040) makes it the best candidate for replacement in the class "0" → class "2" attack scenario.

As in the previous section, we begin by reporting baseline results. Table 4.6 reports the baseline accuracy achieved by each model. All three methods reach solid performance levels, confirming that they handle the core classification task reliably before adversarial perturbations are introduced.

Next, we evaluate the standard adversarial setting, where the goal is to change the model's prediction to *any* incorrect class. Results of these untargeted attacks are shown in Table 4.7.

Particular attention should be given to the XAI subset, which includes samples for which SHAP importance values were precomputed. Since the global ranking used in attacks is averaged over this subset, the results reflect how the method performs when some prior information is available. While results might further improve if importance

Table 4.7: Results of adversarial attacks. The **WordNet-XAI** method replaces impor- tant words with their synonyms, while **WordNet-XAI-ChD** introduces noise by randomly deleting characters from relevant words. The GPT-4o-mini results are based on zero-shot classification. An attack is considered successful if the model's prediction changes from a true positive to any other class.

Attack type	Dataset	Part	BERT Success [%]	GPT-4o-mini Success [%]
WordNet-XAI	AG_News	XAI	4.00	11.00
WordNet-XAI-ChD	AG_News	XAI	2.00	13.00
WordNet-XAI	AG_News	Test	2.27	7.65
WordNet-XAI-ChD	AG_News	Test	2.99	7.69
WordNet-XAI	Wiki_PL	XAI	5.00	5.00
WordNet-XAI-ChD	Wiki_PL	XAI	20.00	12.50
WordNet-XAI	Wiki_PL	Test	1.68	0.28
WordNet-XAI-ChD	Wiki_PL	Test	10.89	0.84

scores were computed per-sample, even this approximation is sufficient to reveal model vulnerabilities, e.g. 1% of AG_News samples and 2.5% of Wiki_PL were successfully attacked using this approach.

We then applied the same global rankings to the full test sets. The attacks proved highly effective overall. The best-performing method involved simply removing characters from the most influential words. A slightly more advanced strategy, replacing words with their synonyms, also yielded solid results.

Interestingly, LLM-based classifiers appear more robust to attacks in the Polish language. This may be related to their high confidence in native classification tasks. However, such robustness should be interpreted cautiously, as it may reflect strong priors learned from training on large multilingual datasets (e.g. Wikipedia).

Tables 4.8 and 4.9 compare the success rates of targeted and nontargeted attacks. This allows us to assess how well our method steers predictions toward a chosen target class. Most scenarios show clear attackability, with the exception of AG_News (A to B), likely due to strong semantic separation between those two classes.

In contrast, for Wiki_PL we expected targeted attacks to succeed more easily, as the class structure is less distinct. However, small edits proved insufficient: the classifier

Table 4.8: Success rates of directed adversarial attacks on a BERT-based classifier. The **WordNet-XAI** method generates adversarial examples by replacing semantically important words with their synonyms using WordNet. The **WordNet-XAI-ChD** method applies additional perturbations by randomly deleting characters from those important words. The notation $A \rightarrow B$ indicates an attempt to intentionally change a sample originally and correctly classified as class A into being misclassified as class B. Mean success rates are reported for each scenario.

Method	Attack type	Dataset	$A \rightarrow B$	$A \rightarrow C$	$A \rightarrow D$	Mean
Targeted	WordNet-XAI	AG_News	2.28	2.34	2.40	2.34
	WordNet-XAI-ChD		2.75	3.45	2.75	2.98
Untargeted	WordNet-XAI		0.94	0.76	0.53	0.74
	WordNet-XAI-ChD		1.46	1.58	0.70	1.25
Targeted	WordNet-XAI	Wiki_PL	4.44	2.22	2.22	2.96
	WordNet-XAI-ChD		6.67	6.67	8.89	7.41
Untargeted	WordNet-XAI		0.00	0.00	1.11	0.37
	WordNet-XAI-ChD		1.11	0.00	1.11	0.74

Table 4.9: Success rates of directed adversarial attacks on GPT-4o-mini. The **WordNet-XAI** method generates adversarial examples by replacing semantically important words with their synonyms using WordNet. The **WordNet-XAI-ChD** method applies additional perturbations by randomly deleting characters from those important words. The notation $A \rightarrow B$ indicates an attempt to intentionally change a sample originally and correctly classified as class A to be misclassified as class B.

Method	Attack type	Dataset	GPT-4o-mini			
			$A \rightarrow B$	$A \rightarrow C$	$A \rightarrow D$	Mean
Targeted	WordNet-XAI	AG_News	3.10	3.04	2.98	3.04
	WordNet-XAI-ChD		2.92	3.39	3.27	3.19
Untargeted	WordNet-XAI		3.16	3.22	1.17	2.52
	WordNet-XAI-ChD		4.09	3.10	1.46	2.88
Targeted	WordNet-XAI	Wiki_PL	0.00	0.00	0.00	0.00
	WordNet-XAI-ChD		0.00	0.00	0.00	0.00
Untargeted	WordNet-XAI		0.00	0.00	0.00	0.00
	WordNet-XAI-ChD		0.00	0.00	0.00	0.00

remained stable unless larger and more noticeable changes were introduced. It is also worth noting that, due to the origin of this dataset (Wikipedia), LLMs may have been exposed to its content in various languages during training, which could contribute to their resilience.

Mitigation

One common approach to mitigating adversarial attacks is to fine-tune models on perturbed inputs. In our experiments, fine-tuning BERT on adversarial examples led to a noticeable improvement in robustness, particularly against synonym-based attacks. However, the model remained vulnerable to character removal strategies, where performance slightly declined. This can be attributed to tokenization mismatches: once a word is distorted at the character level, its token representation often changes entirely, preventing the model from recognizing it as something seen during training.

A more lightweight and cost-efficient alternative may involve detecting perturbations before passing inputs to the model. Pre-processing techniques to flag or correct suspicious modifications, especially character-level noise, could enhance robustness without retraining the full model.

Finally, we tested the effectiveness of fine-tuning BERT on perturbed samples from the *Wiki_PL* dataset. The classifier's accuracy on clean data remained stable, while its resistance to *WordNet-XAI* attacks improved significantly (0.28% success rate), likely due to its exposure to synonym variants during training. In contrast, the model still struggled with *CharDiscard* attacks, showing a 7. 54% success rate and reinforcing the difficulty of defending against character-level manipulations using fine-tuning alone.

Remark 14

Differential ranking steers exact label flips. With few edits an attacker can force class $A \rightarrow B$.

4.3 Summary

Adversarial attacks are not just a threat; they are a practical, targeted audit of model behavior that complements conventional metrics. By probing decisions at their most fragile points, attacks reveal failure modes that accuracy or F1 alone cannot capture. In parallel, explainability (XAI) turns this probing into a controlled experiment: global or local importance scores indicate where to perturb and how to do so with minimal semantic drift. Our results show that XAI-guided variants make attacks faster, more precise, and easier to reproduce, while human evaluations confirm that many adversarial edits preserve perceived meaning.

This dual-use character argues for treating robustness and explainability as coupled

objectives rather than separate tracks. Robust models tend to yield cleaner, more stable explanations; explanations, in turn, enable systematic, black-box stress tests that surface spurious cues and brittle boundaries. In practice, the most informative evaluations are portfolio-based: combine standard classification metrics (e.g. F1, FPR95), OOD screening and cross-dataset generalization, unsupervised structure tests (k-means AMI/ARI and MDS separability), explanation stability/sanity checks, and both plain and XAI-guided adversarial attack success (targeted/untargeted, black-box, budgeted). Doing so embeds adversarial testing naturally within an *audit-measure-improve* loop, offering actionable signals for mitigation (data augmentation, preprocessing, regularization) without overfitting to any single score.

In short, adversarial attacks and XAI are complementary lenses on model trustworthiness. Together, they provide a coherent, lightweight framework for diagnosing weaknesses and steering models toward reliability that matters in real deployments.

Chapter 5

Improving Image Classifier Robustness via Explanation-Guided Fine-Tuning

In this chapter we propose a lightweight, explanation-guided method for improving the robustness of image classifiers without sacrificing headline accuracy. The core idea is to explicitly shape where a model locates its predictive evidence: away from spurious background context and toward object-centric regions of interest (ROIs). The method follows the *audit–measure–improve* discipline introduced earlier: we first diagnose context reliance through quantitative saliency-based evaluation (which also enable us to identify classes requiring improvement), then apply small, targeted fine-tuning losses that bias the model toward object evidence, and finally re-evaluate robustness using complementary metrics, including adversarial stress tests.

While our earlier case studies focused on text, the same audit discipline applies naturally to vision models. In image classification, headline accuracy on ImageNet-style benchmarks can mask brittle, non-robust representations: convolutional networks often rely on backgrounds, co-occurring artifacts, or dataset-specific context rather than on the object itself. As shown by [126], many classes achieve high top-1 accuracy even though their predictions are driven primarily by contextual cues. A particularly clear illustration of this phenomenon is provided by [127]. Figure 5.1 plots per-class accuracy against the class robustness score (CRS) for EfficientNet-B0 on ImageNet. The plot reveals a striking failure mode: a substantial subset of classes achieves very high classification accuracy while exhibiting low robustness scores. These classes are recognized correctly, yet for the wrong reasons.

Following this analysis, we adopt a compact saliency-based diagnostic, the *robustness score*, defined as the fraction of attribution mass that falls within the ground-truth region of interest (ROI). Low values indicate reliance on background evidence even when classification accuracy is high. This score provides an effective audit signal that motivates targeted, representation-level interventions rather than wholesale retraining.

Formally, following [126], let $\phi(I, x, y) \in [0, 1]$ be a saliency map for image I and let $\mathrm{bbox}(I, x, y) \in \{0, 1\}$ indicate whether pixel (x, y) lies inside the ground-truth box for the class of I. The per-image robustness score is

$$\mathrm{rs}(I) = \frac{\sum_{(x,y)\in I} \mathrm{bbox}(I, x, y) \cdot \phi(I, x, y)}{\sum_{(x,y)\in I} \phi(I, x, y)}. \tag{5.1}$$

The per-class score averages $\mathrm{rs}(I)$ over the validation images of class c:

$$\mathrm{crs}(c) = \frac{1}{|\{I : \ell(I) = c\}|} \sum_{I:\ell(I)=c} \mathrm{rs}(I). \tag{5.2}$$

Using ImageNet CLS-LOC with a ResNet-152 backbone and Grad-CAM++, [126] rank classes by the class robustness score and show that many categories combine high top-1 accuracy with low localization-aligned evidence. In such classes, saliency mass frequently lies on backgrounds or co-occurring artifacts rather than on the object itself. A representative case study (e.g. *rugby ball*) shows saliency concentrated on players, and masking the ball does not appreciably reduce confidence, revealing strong context reliance.

The authors also describe two "natural" adversarial failure modes enabled by spurious cues: (i) presenting an object outside its usual context leads to misses, and (ii) showing only the context (without the object) triggers highly confident misclassifications into context linked classes. Beyond model behavior, the audit surfaces dataset issues that depress robustness: missing or too-tight boxes for small objects (e.g. balls, insects) and class polysemy (e.g. *volleyball* as sport vs. as ball), where label, box, and content are misaligned. Crucially, the class-level rankings are stable across architectures (AlexNet, ResNet, EfficientNet variants) and across saliency methods (Grad-CAM++, Smooth Grad-CAM++). The same low-score categories reappear repeatedly, indicating that the phenomenon is driven by data correlations rather than a single model or explanation. Similar patterns are reported beyond ImageNet (e.g. Pascal VOC), where categories

like *bottle*, *chair*, or *boat* show saliency anchored in typical contexts (tables/people, indoor scenes, water). In general, low robustness scores reliably flag classes whose predictions hinge on context or artifacts, offering actionable signals for data curation and robustness-oriented model refinement.

We believe that their robustness score is a useful addition to our portfolio of audits, but it is not a silver bullet. Context can be legitimately predictive (e.g. *boat* with water, sports gear with field lines), so evidence just outside the box is sometimes helpful. Thus, very low class scores almost surely flag shortcut use, yet *extremely* high scores can also be a warning: the model may overfit to box interiors and ignore informative context. Noisy or incomplete boxes further bias the estimate. Use this score in concert with accuracy, OOD screening, geometry and clustering checks, and adversarial stress tests to maintain a balanced view of robustness.

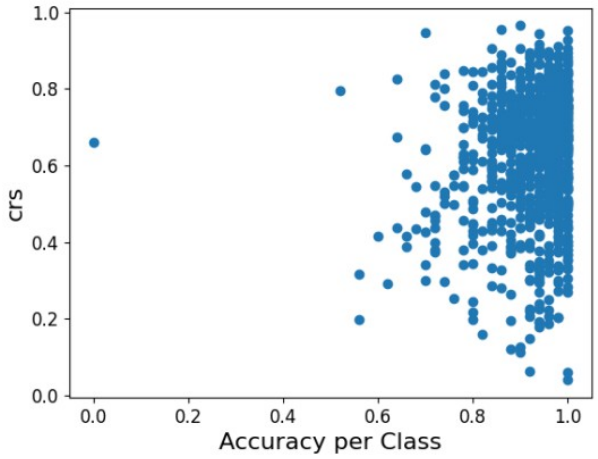


Figure 5.1: Accuracy per class versus class robustness score (CRS) for EfficientNet-B0 on ImageNet, adapted from [127]. Each point corresponds to one class. The bottom-left region highlights classes that achieve high accuracy despite low robustness scores, indicating that predictions rely on background or contextual cues rather than object evidence. This discrepancy motivates explanation-based audits and representation-level interventions beyond headline accuracy.

5.1 A Lightweight, Explanation-Guided Method for Object-Centric Robustness

We start from a trained, high-accuracy model audited with the robustness score. We propose a lightweight, explanation-guided fine-tuning method that explicitly aims to *shift the predictive evidence* from background context toward the ground-truth regions of interest (ROIs). To this end, we fine-tune the model with a composite loss that leverages bounding boxes and, optionally, CAM attributions as weak guidance. In the following, we define the individual components and the final objective.

Box-aware classification. Given a preprocessed image x and the union ROI mapped to the same coordinates, we extract the object crop, resize it to 224×224, and apply cross-entropy on the crop:

$$\mathcal{L}_{\text{crop}} \;=\; \text{CE}\Big(f\big(\text{Resize}(\text{Crop}(x;\text{ROI}))\big),\, y\Big). \tag{5.3}$$

This encourages correct predictions from *object evidence alone*.

Background debiasing. We mask the ROI on the full image to obtain $x_{\text{bg masked}}$ (either zeroing or adding light noise inside the box) and penalize overconfidence on the background:

$$\mathcal{L}_{\text{bg}} \;=\; \max\big(p_\theta(y \mid x_{\text{bg masked}}) - m,\, 0\big), \quad m \in [0, 1). \tag{5.4}$$

High values indicate reliance on context when the object is absent.

CAM-alignment. Let $S \in [0,1]^{H_c \times W_c}$ be a min-max normalized Grad-CAM++ map for the true or predicted class, and let $M \in \{0,1\}^{H_c \times W_c}$ be the ROI mask projected to the CAM grid. We treat S as a detached signal and use two priors:

$$\mathcal{L}_{\text{out}} = \frac{1}{H_c W_c} \sum_{u,v} (1 - M_{uv})\, S_{uv}, \qquad \text{(penalize saliency outside ROI)}, \tag{5.5}$$

$$\mathcal{L}_{\text{IoU}} = 1 \;-\; \frac{\langle M, S \rangle}{\|M\|_1 + \|S\|_1 - \langle M, S \rangle + \varepsilon}, \qquad \text{(maximize CAM/ROI overlap)}. \tag{5.6}$$

These terms gently steer attention into the ROI without dominating training.

Final objective. The fine-tuning loss combines the standard cross-entropy on the full image with the terms above:

$$\mathcal{L} = \underbrace{\mathrm{CE}\big(f(x), y\big)}_{\text{full image}} + \alpha\,\mathcal{L}_{\text{crop}} \;+\; \beta\,\mathcal{L}_{\text{bg}} \;+\; \gamma_{\text{out}}\,\mathcal{L}_{\text{out}} \;+\; \gamma_{\text{IoU}}\,\mathcal{L}_{\text{IoU}}. \tag{5.7}$$

The weights $\alpha, \beta, \gamma_{\text{out}}, \gamma_{\text{IoU}} \geq 0$ control the strength of each component.

5.1.1 Experimental Setup

We first rank ImageNet classes by the per-class robustness score $\mathrm{crs}(c)$. We then select the 20 lowest-scoring (worst) classes and fine-tune a pretrained ResNet–152 on the corresponding training images, evaluating accuracy and robustness on the validation split.

We conducted a partial hyperparameter search over all auxiliary loss weights, i.e. α (crop loss), β (background penalty), γ_{out} (outside–ROI CAM penalty) and γ_{IoU} (CAM/ROI IoU prior), using Optuna [128]. The search was driven by a gated multi objective favoring robustness without sacrificing accuracy:

$$\max_{\theta}\ \mathrm{rs}(\theta) \quad \text{s.t.} \quad \mathrm{acc}(\theta) \geq \tau, \qquad \tau = 0.80. \tag{5.8}$$

Trials with acc $< \tau$ were discarded; among feasible trials we picked the highest rs (ties broken by accuracy).

Despite searching the space, the best configuration was *sparse*: only the background and outside-ROI terms received non-zero weights. Concretely, the optimum was used $\beta = 0.30$ (Eq. 5.4) and $\gamma_{\text{out}} = 0.15$ (Eq. 5.5), with all other weights set to 0.

Intuitively, additional priors failed the accuracy gate or did not improve robustness once background reliance was penalized and saliency was gently nudged into the ROI. The selected setting shifts evidence toward the object while preserving core task performance.

5.1.2 Results

Table 5.1 summarizes per-class changes after fine-tuning the 20 lowest-robustness ImageNet classes. On average, robustness (CAM mass inside the ground-truth ROI) improves by +0.071 (from 0.212 to 0.283), while top-1 accuracy drops by only -0.01

(from 0.928 to 0.918). Put simply: we shift evidence toward the object by ~7 percentage points at a cost of about 1 percentage point in accuracy.

Accuracy remains unchanged or improves in 12/20 classes and degrades in 8/20. The greatest drops occur for *horizontal_bar* (-0.10), *space_bar* (-0.08), *volleyball* (-0.06) and *bearskin* (-0.06). Despite these, robustness still rises in *19/20* categories; the only non-gain is *space_bar* (slight -0.002). Most of those classes are among those that the authors of [126] identified as problematic from the perspective of data set issues.

The class "sunglasses" shows a clear win: robustness +0.126 with a small increase in accuracy (+0.02). Qualitatively, Grad-CAM++ heatmaps move off background distractors (e.g. a water bottle) and onto the glasses frame; see Figure 5.2. *Switch* also improves on both axes (+0.02 Acc, +0.081 Rob), with attention focusing on the toggle/rocker rather than the surrounding wall; see Fig. 5.3. Sports balls (*basketball*, *rugby_ball*, *volleyball*, *ping-pong_ball*) gain robustness (+0.027 to +0.084) while maintaining or only slightly trading accuracy, indicating reduced reliance on contextual background.

By contrast, *space_bar* is challenging. Accuracy drops (-0.08) and robustness is flat (-0.002). Visualizations suggest the CAM, which previously lit the whole keyboard, now fragments over key clusters or the desk surface, missing the narrow ROI; see Figure 5.4. A likely cause is semantic/visual overlap with *horizontal_bar*, which itself loses accuracy (-0.10) even though its robustness increases (+0.049). This pair illustrates a limit of box-guided priors for thin, elongated structures and near-class confusions.

5.1.3 Black-Box Adversarial Stress Testing

We also probe robustness with adversarial examples. Under the white-box *AutoAttack* suite [83] (an ensemble that includes APGD-CE, APGD-T, FAB, and Square), both the baseline and our fine-tuned model fail at standard L_∞ budgets: this is expected, because we never trained with adversarial objectives or regularizers explicitly targeting gradient-based attacks. Established defenses that do improve white-box robustness include adversarial training [89], TRADES [90], and techniques like AugMix or randomized smoothing for complementary gains [91], [130]. Our goal here is different: by re-centering evidence on the object, we expect stronger resistance to *black-box*, score-based attacks that operate via localized, image-space perturbations. Accordingly, adversarial

Table 5.1: Baseline versus fine-tuned models on the 20 classes with the lowest robustness scores. We report top-1 accuracy (Acc) and robustness (Rob), defined as the fraction of Grad-CAM mass inside the ground-truth ROI. Deltas correspond to fine-tuned minus baseline performance.

Class	Acc (base)	Rob (base)	Acc (ft)	Rob (ft)	ΔAcc	ΔRob
sunglasses	0.84	0.255	0.86	0.381	0.02	0.126
snorkel	0.98	0.282	0.98	0.406	0.00	0.124
bathing_cap	0.90	0.192	0.86	0.309	-0.04	0.117
diaper	0.88	0.277	0.88	0.391	0.00	0.114
miniskirt	0.88	0.261	0.90	0.371	0.02	0.110
swimming_trunks	0.86	0.186	0.88	0.286	0.02	0.101
bearskin	1.00	0.217	0.94	0.306	-0.06	0.089
croquet_ball	0.98	0.247	0.96	0.331	-0.02	0.084
basketball	0.96	0.061	0.96	0.145	0.00	0.084
rugby_ball	0.90	0.241	0.94	0.324	0.04	0.083
switch	0.94	0.278	0.96	0.358	0.02	0.081
volleyball	1.00	0.088	0.94	0.165	-0.06	0.077
racket	0.90	0.258	0.88	0.328	-0.02	0.070
flagpole	0.98	0.279	0.94	0.340	-0.04	0.061
horizontal_bar	0.92	0.214	0.82	0.263	-0.10	0.049
ping-pong_ball	0.98	0.085	0.98	0.112	0.00	0.027
balance_beam	0.80	0.213	0.84	0.235	0.04	0.023
puck	0.98	0.276	0.98	0.280	0.00	0.004
pickelhaube	0.88	0.130	0.94	0.131	0.06	0.001
space_bar	1.00	0.191	0.92	0.189	-0.08	-0.002
Mean	**0.928**	**0.212**	**0.918**	**0.283**	**–0.01**	**0.071**

Table 5.2: Baseline versus fine-tuned model under the Square black-box attack [129]. We report clean accuracy, accuracy under attack, and conditional robust accuracy computed on clean-correct samples. Each row averages results over 1000 images.

	Clean acc.	Acc. under attack	Robust acc. (clean-correct)
$\varepsilon = 2/255$	0.928 → 0.918 (−0.010)	0.429 → 0.527 (+0.098)	0.462 → 0.574 (+0.112)
$\varepsilon = 4/255$	0.928 → 0.918 (−0.010)	0.164 → 0.222 (+0.058)	0.177 → 0.242 (+0.065)
$\varepsilon = 8/255$	0.928 → 0.918 (−0.010)	0.041 → 0.051 (+0.010)	0.044 → 0.056 (+0.012)

evaluation is used here not as a training objective, but as an external stress test that probes whether changes in evidence localization translate into practical robustness gains.

Concretely, we adopt *Square Attack* [129], a query-efficient black-box method that

never crops or zooms the image. In the L_∞ setting, the perturbation constraint $\|\delta\|_\infty \leq \varepsilon$ means that each pixel may change by at most ε in absolute value; the feasible region is thus the hypercube $[-\varepsilon, \varepsilon]^d$ centered at the image x. The "sphere" of radius ε corresponds to its boundary, i.e., points satisfying $\max_i |\delta_i| = \varepsilon$.

Importantly, Square Attack operates directly with *maximal* perturbation steps: pixel values are flipped by $\pm\varepsilon$ within selected square patches, rather than being gradually adjusted. This binary update rule makes the perturbation discrete and maximally informative under the query budget.

The algorithm initializes with a random δ_0 on the boundary and then iteratively modifies δ inside randomly placed square patches. The patch size follows a schedule that gradually decreases over time. After each update, the perturbed image $x + \delta$ is clipped to the valid pixel range, ensuring $\|\delta\|_\infty \leq \varepsilon$. A candidate update is accepted only if it decreases the classification margin (or increases the loss); otherwise, it is discarded. The process repeats until the query budget is exhausted.

Intuitively, spatially coherent patch updates allow the attack to quickly probe vulnerable regions of the image while remaining fully agnostic to model gradients (and therefore robust to gradient masking).

Table 5.2 reports clean accuracy, accuracy under attack, and a robust accuracy computed on clean-correct samples, which measures the fraction of originally correct predictions that remain correct under attack. Across all L_∞ budgets, the fine-tuned model consistently outperforms the baseline in black-box robustness, with only a minor drop in clean accuracy (about 1 pp). For example, *accuracy under attack* improves by +9.8 pp at $\varepsilon = 2/255$, +5.8 pp at $\varepsilon = 4/255$, and +1.0 pp at $\varepsilon = 8/255$. The stricter *robust accuracy* shows a similar trend (+11.2, +6.5, and +1.2 pp, respectively). This aligns with our CAM-based audit: when model evidence is concentrated on the object rather than the background, localized patch perturbations are less effective in flipping the prediction.

Figure 5.5 illustrates representative adversarial examples. The perturbations are imperceptible to the human eye, yet they reliably fool the baseline model while the fine-tuned model remains robust.

In summary, CAM-guided fine-tuning does not harden the model against powerful white-box attacks (by design), but it does improve resilience to realistic black-box, patchy perturbations at fixed L_∞ budgets, complementing our portfolio of audits beyond headline accuracy.

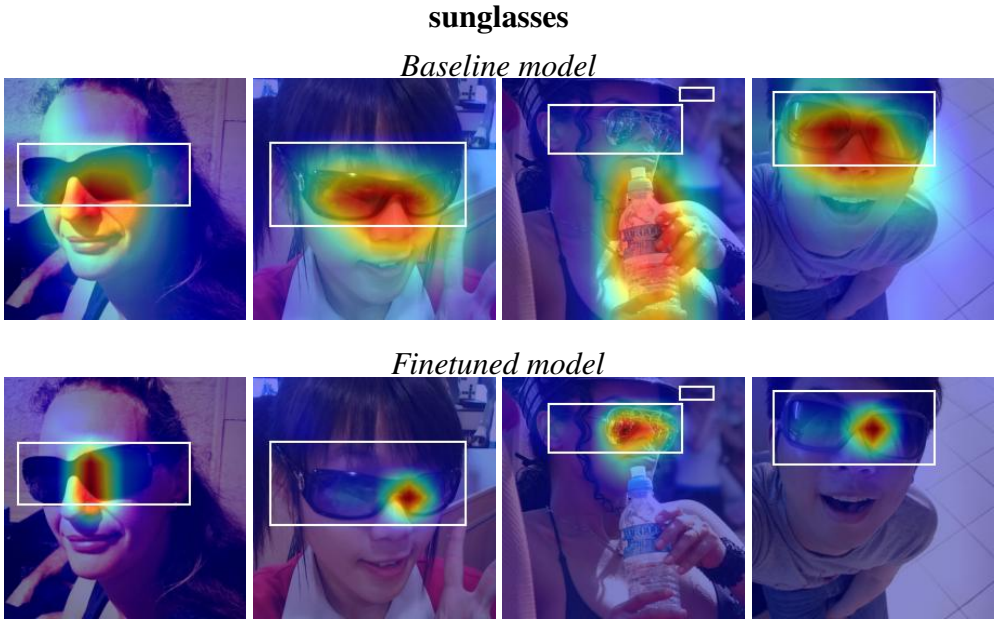


Figure 5.2: Comparison of the Baseline model and the Finetuned model for the "sunglasses" class, highlighting a strong improvement in the robustness score after fine-tuning.

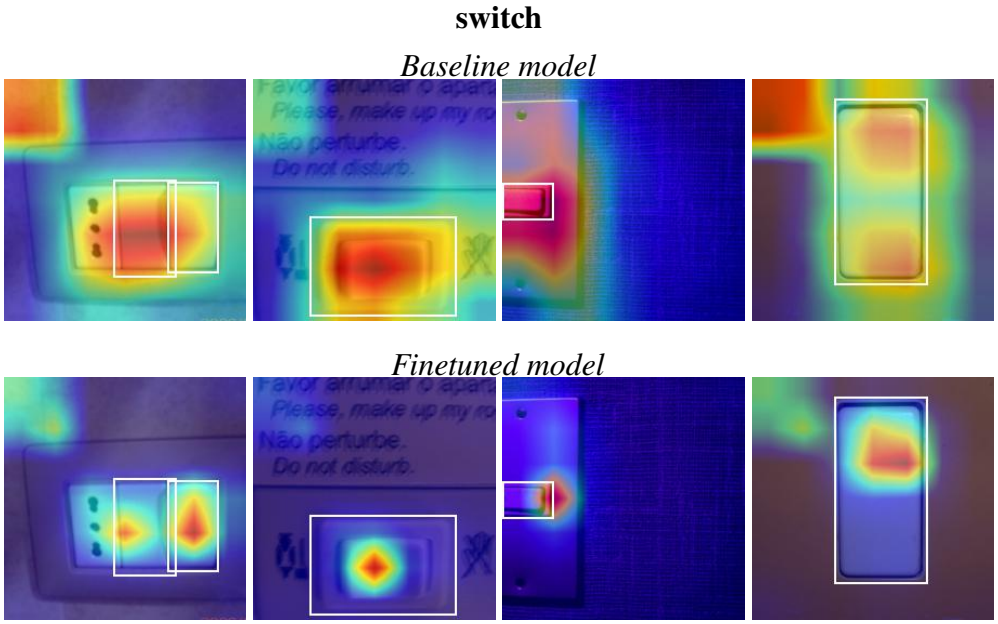


Figure 5.3: Comparison of the Baseline model and the Finetuned model for the "switch" class, showing a medium improvement in the robustness score after fine-tuning.

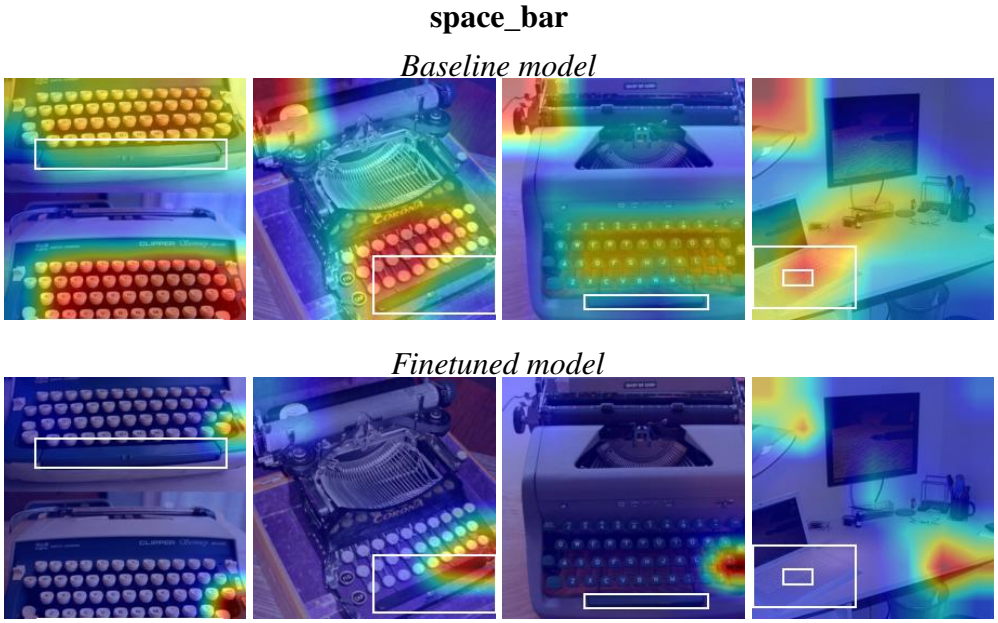


Figure 5.4: Comparison of the Baseline model and the Finetuned model for the "space_bar" class, showing no improvement in the robustness score after fine-tuning.

Figure 5.5: Examples of transformed images (the perturbation is hardly perceptible) that successfully deceive the baseline model but not the fine-tuned one. In the second row we show difference heatmaps (averaged across channels) relative to the original image. White indicates a positive shift of $+4/255$, black denotes a negative shift of $-4/255$, and gray corresponds to unchanged pixels. Thus, only three discrete intensity levels appear in the heatmap.

5.2 Summary

This chapter extended our audit-measure-improve loop from text to vision, showing that the same disciplined workflow applies across modalities. We adopted the robustness score of [126] as a lightweight audit: saliency mass should fall inside the ground-truth ROI, not drift into background shortcuts. Using Grad-CAM/Grad-CAM++ as our attribution backbone, we identified the 20 weakest ImageNet classes under this audit and treated them as targets for refinement.

We then fine-tuned a strong baseline with a minimal, box-aware objective that requires confidence from object evidence and discourages background reliance. A small background penalty plus a light outside-ROI saliency penalty yielded the best trade-off in a gated search (maximize robustness subject to acc $\geq 80\%$). On average, class-wise robustness improved by about +0.07 while top-1 accuracy changed only marginally (~ -0.01). Qualitatively, CAMs re-centered on objects: for *sunglasses*, attention moved off spurious context (e.g. water bottles) and into the lenses and frame (see Fig. 5.2); *switch* likewise shows cleaner evidence (Fig. 5.3). Some categories remain challenging: *space_bar* vs. visually similar *horizontal_bar* illustrates label-set ambiguity, where CAMs split attention between keyboard parts and desk regions (Fig. 5.4).

We stress that the robustness score is informative but not absolute. Very low values reliably flag context overuse; however, "maximal" scores are not always desirable either, as legitimate context can carry signal. The right target is object-anchored evidence with measured use of scene cues, not saliency that is artificially constrained to boxes.

Finally, we probed adversarial robustness. Unsurprisingly, neither baseline nor fine-tuned models withstand strong white-box attacks such as AutoAttack [83] – we did not train with gradient-level defenses. Yet under black-box, score-based Square Attack [129] the fine-tuned model is consistently more robust across L_∞ budgets (Table 5.2), in line with its saliency-driven shift toward object evidence. This supports our central thesis: targeted, explanation-guided fine-tuning can make embeddings cleaner and decisions more accountable without chasing headline accuracy.

Overall, the image results mirror our findings in HTTP/URL and text: start with transparent audits (here, ROI-aligned saliency), intervene with small, well-motivated losses, and validate with a *portfolio* of tests-standard accuracy, attribution quality, and adversarial stress. The same pattern scales to new domains, encourages human-in-the-loop inspection, and produces models that fail more gracefully when they fail.

Chapter 6

Final Conclusions

This chapter closes the loop opened in the introduction. There, we argued that trustworthy learning systems do not arise from accuracy alone but from a disciplined *audit* → *measure* → *improve* cycle that makes evidence use, representation geometry, and boundary behavior observable and correctable. The central hypothesis of this dissertation was that such a cycle can not only reveal failure modes, but also enable measurable and targeted improvements in model robustness and reliability. Here, we gather the threads, state clearly what worked and why, acknowledge limits, and apply a pragmatic ethics lens to deployment, emphasizing trustworthiness as a core requirement – namely, that models should not rely on users or society to uncover failures that could have been anticipated through systematic evaluation. The result is a compact operating picture: a model is reliable when we can see how it succeeds and detect how it fails. Doing so requires subjecting the model to a portfolio of tests and, when necessary, pursuing targeted improvements that those same tests can verify.

Across modalities, the pattern was consistent. Explanations served as an *audit* that surfaced shortcuts and brittle cues even when the test accuracy looked healthy. We operationalized this with a portfolio of established tests from the literature – attribution sanity checks, clustering metrics, and OOD fitness – and *added* custom procedures where gaps remained. Representation diagnostics *measured* whether the latent space supported robust behavior rather than merely memorized surfaces. On the improvement side, we introduced lightweight, *custom fine-tuning*: a contrastive refresh for text and a localization-aware, box-guided tune for vision, both of which *improved* geometry and stress performance at minimal cost. The practical lesson is less about a single trick and

more about a workflow: inspect the evidence, read the space, and only then change the model in ways that those same audits can verify. When these elements are kept in view, reliability becomes easier to achieve and maintain.

As a complementary *audit*, we also employed *adversarial evaluation*. In text, we combined established attacks with our XAI-based method. In vision, we used black-box, patch-style stressor to check whether evidence remained object-anchored. These probes exposed brittle cues that standard classification performance metrics would miss.

Taken together, these elements turn reliability from an abstract notion into an operational discipline: we *see* (audit) what evidence the model uses, *measure* whether its representation space supports robust decisions, and *improve* it with small, targeted steps that our own audits can verify. To facilitate systematic interaction with trained models in practice, we distill the proposed *audit–measure-improve* framework into an operational guide. The guidance is intended to support both routine model analysis and targeted improvements by translating the framework's principles into concrete, repeatable steps.

Guidance Before Open-World Deployment for Systematic Model Preparation

Audit (see what the model uses).

1. Establish a *reference snapshot*: a stable test set (with embeddings, attributions, and metrics) against which future models will be compared.

2. Examine classifier decisions using *global* explainability (aggregated statistics or averages) and *local* inspection of individual samples (manual review of representative and edge cases).

3. Expand the *testing portfolio* with concept drift in mind: add OOD detection and generalization checks, clustering and embedding visualizations, adversarial attacks, and any domain-appropriate probes (e.g. Class Robustness Score for images).

4. Recall that the *reference snapshot* may evolve as new data arrive; when it changes, rerun the full procedure and re-establish thresholds and baselines.

Measure (check if the space supports robust decisions).

1. From the expanded audit portfolio, *run all tests* to obtain a multi-view assessment of performance across tasks (classification, OOD, clustering, visualization-based sanity checks, adversarial examples).

2. Compare the results with the reference snapshot, previous model versions, and explicit target expectations for this release.

3. Track *representation health*: silhouette and between/within variance; add ARI/AMI when labels exist; monitor kNN purity to catch boundary fraying.

4. Evaluate *OOD fitness* with robust metrics: AUROC, AUPR, and operational FPR@95%. Moreover, check the calibration of your model.

5. *Curate OOD sets*, even extreme or synthetic ones ("from outer space" domains), to probe generalization limits; mix near-shift and far-shift sources.

6. Watch *input drift* on key features.

7. Human evaluation, although expensive, is always a good idea.

Improve (turn findings into safe, reversible actions).

1. *Design fixes from metrics.* Start by mapping each degraded metric to a concrete hypothesis and a minimal intervention. Describe the expected effect and how you will verify it.

2. *Be standardized and methodical.* Use a fixed template for experiments (goal, change, datasets, metrics, stop rules), version every artifact, and enable quick rollback to the last "good" release.

3. *Prefer light levers first.* Not every issue requires retraining. Try calibration, threshold tuning, or an auxiliary module (e.g. OOD-based gating or abstention) before touching the backbone.

4. *If you finetune, do it carefully.* Plan the change: collect data (new vs. replay), pick loss and decide if any architecture tweaks are needed. Use small learning rates, warmup, early stopping, and regularization to avoid *catastrophic forgetting*. Expect that eval may dip in early epochs.

5. *Monitor during training.* Track a subset of *measure* signals online (e.g. kNN purity, OOD AUROC/TNR@95) across epochs and key data; stop when gains plateau or any protected slice regresses.

6. *Validate before promote.* Compare against the pinned reference on all core metrics; promote only with no red regressions and documented trade-offs.

7. *Document and fix the baseline.* Record what changed, why it helped, and how to revert. Define a new reference snapshot (test set, thresholds, calibration) if you decide to advance the baseline.

Together, these steps show how a developer can move from a single trained model to a monitored and improvable system. The goal is not to chase a marginal gain on a single benchmark metric, but to cultivate a habit of inspection, measurement, and controlled improvement. In contrast to much of the literature, where progress is equated with a one-point improvement in the F1-score or accuracy, we argue that the real novelty lies in showing that models can be improved more thoroughly, monitored over time, and adapted responsibly to their true operating conditions. High scores on a leaderboard are a good starting point, but they tell us little about which model will serve us best in practice. The results presented throughout this dissertation support the central hypothesis that trustworthiness is not an emergent property of scale or accuracy alone, but a consequence of systematic inspection, measurement, and targeted improvement applied over the lifecycle of a model.

6.1 Limitations, ethics, and future directions

Every approach has its boundaries, and the framework proposed in this dissertation is no exception. Several limitations deserve explicit mention. Some of the tests that we advocate can be computationally expensive, especially when validation is run during training or across many slices. Our focus was deliberately narrow: we emphasized the separability of the embedding space and the improvement of specific metrics while keeping accuracy intact. This is not a universal recipe. Different applications will value different objectives, though we argue that OOD robustness is fundamental in almost every case. Moreover, while adversarial attacks served us well as audits, and our finetuning helped mitigate some of their effects, hardening requires more systematic

defenses. Simple data augmentation by mutating samples is one option; more advanced strategies include adversarial training, TRADES, randomized smoothing, or monitoring additional metrics such as certified radius or robust accuracy curves. Finally, the tests that we introduced continuously reveal weaknesses that can be addressed, but they are not a cure for all: no portfolio of checks can anticipate every possible failure, and residual risk must be acknowledged.

These limitations directly connect to the ethical dimension. One of the most important concerns is the release of models or research that have not been tested from multiple angles. As a community, we should move towards more standardized evaluations and structured reporting of results, so that performance claims are not tied to a single number but to a richer profile of behavior. Investing in ethical AI is no longer optional: machine learning systems intersect with our daily lives in an ever wider range of situations, and failures quickly spill into society. Adversarial attacks illustrate this tension. They are tools used by researchers to probe and strengthen systems, but in practice they are also the techniques of attackers who seek to subvert them. Studying adversarial robustness is, therefore, both a scientific and an ethical necessity. It matters what information an adversary has (e.g., access to weights versus only outputs), but it is equally important to remember that a harder attack is not impossible. We can use this asymmetry to our advantage: the more we prepare, the higher the cost to the attacker, and the more resilient the system in deployment.

Looking ahead, there are several directions in which this work can be extended. First, the agenda is too large for one researcher alone; what is needed is a shift in the community mindset. Benchmarking and standardization across modalities and problem classes should become a norm, and reviewers should expect to see not just a leaderboard score, but a portfolio of tests. Second, research should focus more squarely on methods that improve *OOD generalization*, as in-distribution generalization is largely a solved problem. Third, the methodology can and should be adapted to new data types, for example, extending the loop of audit, measure, and improve to graph-structured data, with a special emphasis on explainability in relational domains. Fourth, we must confront the stability of the explainability methods themselves. SHAP, for example, is widely used but computationally heavy; moreover, recent studies demonstrate that explanation methods are themselves vulnerable to attacks. If our audits depend on explanations, we need to trust them, and developing more robust, efficient, and reliable XAI techniques is a critical research frontier.

Together, these methodological, ethical, and forward-looking lessons point to a broader conclusion. Progress in machine learning cannot be measured solely by squeezing out one more percentage point of F1-score. This paradigm, which is still dominant in the literature, is too narrow to guaranty reliability in practice. This dissertation has argued and demonstrated that models can be tested more thoroughly, monitored over time, and improved responsibly with lightweight but principled updates. The high benchmark accuracy remains a valuable starting point, but it tells us little about which model will actually serve us best in a real application. By embedding models into a cycle of *audit, measure, and improve*, we move closer to systems that fail less often, fail more gracefully, and exhibit increased reliability in critical deployment contexts.

Bibliography

[1] M. W. Spratling, *A comprehensive assessment benchmark for rigorously evaluating deep learning image classifiers*, 2025.

[2] R. Geirhos, L. Temme, J. Rauber, B. Schölkopf, and F. A. Wichmann, "Shortcut learning in deep neural networks," *Nature Machine Intelligence*, 2020.

[3] F. Croce, Y. He, and M. Hein, "Robustbench: A standardized benchmark for adversarial robustness," *arXiv preprint*, 2021.

[4] A. Nguyen, J. Yosinski, and J. Clune, "Deep neural networks are easily fooled: High confidence predictions for unrecognizable images," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[5] R. Jia and P. Liang, "Adversarial examples for evaluating reading comprehension systems," *arXiv preprint arXiv:1707.07328*, 2017.

[6] M. T. Ribeiro, T. Wu, C. Guestrin, and S. Singh, "Beyond accuracy: Behavioral testing of nlp models with checklist," *arXiv preprint arXiv:2005.04118*, 2020.

[7] J. R. Zech, M. A. Badgeley, M. Liu, A. B. Costa, J. J. Titano, and E. K. Oermann, "Variable generalization performance of a deep learning model to detect pneumonia in chest radiographs," *PLoS Medicine*, vol. 15, no. 11, e1002683, 2018.

[8] L. Oakden-Rayner, J. Dunnmon, G. Carneiro, and C. Ré, "Hidden stratification causes clinically meaningful failures in machine learning for medical imaging," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, 2020, pp. 1513–1520.

[9] C. J. Kelly, A. Karthikesalingam, M. Suleyman, G. Corrado, and D. King, "Key challenges for delivering clinical impact with artificial intelligence," *BMC Medicine*, vol. 17, no. 1, p. 195, 2019.

[10] R. Michelmore, M. Kwiatkowska, and Y. Gal, "Evaluating uncertainty quantification in end-to-end autonomous driving control," in *Proceedings of the Conference on Robot Learning (CoRL)*, 2018.

[11] D. Amodei, C. Olah, J. Steinhardt, P. Christiano, J. Schulman, and D. Mané, "Concrete problems in ai safety," *arXiv preprint*, 2016.

[12] R. Bommasani *et al.*, "On the opportunities and risks of foundation models," *arXiv preprint*, 2021.

[13] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *arXiv preprint arXiv:1312.6199*, 2013.

[14] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.

[15] D. Hendrycks and K. Gimpel, "A baseline for detecting misclassified and out-of-distribution examples in neural networks," *arXiv preprint arXiv:1610.02136*, 2016.

[16] K. Lee, K. Lee, H. Lee, and J. Shin, "A simple unified framework for detecting out-of-distribution samples and adversarial attacks," *Advances in neural information processing systems*, vol. 31, 2018.

[17] K. Szyc, T. Walkowiak, and H. Maciejewski, "Why out-of-distribution detection experiments are not reliable-subtle experimental details muddle the ood detector rankings," in *Uncertainty in Artificial Intelligence*, PMLR, 2023, pp. 2078–2088.

[18] "Artificial intelligence risk management framework (ai rmf 1.0)," National Institute of Standards and Technology, Gaithersburg, MD, Tech. Rep. NIST AI 100-1, Jan. 2023, Version 1.0.

[19] *Regulation (eu) 2024/1689 of the european parliament and of the council of 13 june 2024 laying down harmonised rules on artificial intelligence (artificial intelligence act)*, Official Journal of the European Union, ELI: 32024R1689, 2024.

[20] Center for AI Safety, *Statement on ai risk*, https://www.safe.ai/statement-on-ai-risk, One-sentence statement on catastrophic AI risk; signatories include Geoffrey Hinton, Yoshua Bengio, Daniel Kahneman, and others, 2023.

[21] Y. Bengio, S. Mindermann, D. Privitera, T. Besiroglu, R. Bommasani, S. Casper, Y. Choi, D. Goldfarb, H. Heidari, L. Khalatbari, *et al.*, "International scientific report on the safety of advanced ai (interim report)," *arXiv preprint arXiv:2412.05282*, 2024.

[22] Y. Bengio, G. Hinton, A. Yao, D. Song, P. Abbeel, T. Darrell, Y. N. Harari, Y.-Q. Zhang, L. Xue, S. Shalev-Shwartz, *et al.*, "Managing extreme ai risks amid rapid progress," *Science*, vol. 384, no. 6698, pp. 842–845, 2024.

[23] C. Sammut and G. I. Webb, *Encyclopedia of machine learning*. Springer Science & Business Media, 2011.

[24] D. R. Cox, "The regression analysis of binary sequences," *Journal of the Royal Statistical Society Series B: Statistical Methodology*, vol. 20, no. 2, pp. 215–232, 1958.

[25] W.-Y. Loh, "Classification and regression trees," *Wiley interdisciplinary reviews: data mining and knowledge discovery*, vol. 1, no. 1, pp. 14–23, 2011.

[26] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[27] F. Rosenblatt, "The perceptron: A probabilistic model for information storage and organization in the brain.," *Psychological review*, vol. 65, no. 6, p. 386, 1958.

[28] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[29] G. G. Chowdhury, *Introduction to modern information retrieval*. Facet publishing, 2010.

[30] K. Sparck Jones, "A statistical interpretation of term specificity and its application in retrieval," *Journal of documentation*, vol. 28, no. 1, pp. 11–21, 1972.

[31] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[32] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *Transactions of the association for computational linguistics*, vol. 5, pp. 135–146, 2017.

[33] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[34] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, K. Erk and N. A. Smith, Eds., Berlin, Germany: Association for Computational Linguistics, Aug. 2016, pp. 1715–1725.

[35] X. Song, A. Salcianu, Y. Song, D. Dopson, and D. Zhou, "Fast wordpiece tokenization," *arXiv preprint arXiv:2012.15524*, 2020.

[36] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[37] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, 2019, pp. 4171–4186.

[38] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," *arXiv preprint arXiv:1907.11692*, 2019.

[39] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[40] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, Ieee, vol. 1, 2005, pp. 886–893.

[41] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 2002.

[42] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *European conference on computer vision*, Springer, 2016, pp. 630–645.

[43] C. C. Aggarwal, A. Hinneburg, and D. A. Keim, "On the surprising behavior of distance metrics in high dimensional space," in *International conference on database theory*, Springer, 2001, pp. 420–434.

[44] S. Lloyd, "Least squares quantization in pcm," *IEEE transactions on information theory*, vol. 28, no. 2, pp. 129–137, 1982.

[45] D. Arthur and S. Vassilvitskii, "K-means++: The advantages of careful seeding," Stanford, Tech. Rep., 2006.

[46] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *kdd*, vol. 96, 1996, pp. 226–231.

[47] E. Schubert, J. Sander, M. Ester, H. P. Kriegel, and X. Xu, "Dbscan revisited, revisited: Why and how you should (still) use dbscan," *ACM Transactions on Database Systems (TODS)*, vol. 42, no. 3, pp. 1–21, 2017.

[48] J. H. Ward Jr, "Hierarchical grouping to optimize an objective function," *Journal of the American statistical association*, vol. 58, no. 301, pp. 236–244, 1963.

[49] F. Murtagh and P. Legendre, "Ward's hierarchical agglomerative clustering method: Which algorithms implement ward's criterion?" *Journal of classification*, vol. 31, no. 3, pp. 274–295, 2014.

[50] S. Raschka and V. Mirjalili, *Python Machine Learning, 3rd Ed.* 3rd ed. Birmingham, UK: Packt Publishing, 2019.

[51] P. J. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," *Journal of computational and applied mathematics*, vol. 20, pp. 53–65, 1987.

[52] L. Hubert and P. Arabie, "Comparing partitions," *Journal of classification*, vol. 2, no. 1, pp. 193–218, 1985.

[53] N. X. Vinh, J. Epps, and J. Bailey, "Information theoretic measures for clusterings comparison: Is a correction for chance necessary?" In *Proceedings of the 26th annual international conference on machine learning*, 2009, pp. 1073–1080.

[54] P. C. Mahalanobis, "On the generalized distance in statistics," *Sankhyā: The Indian Journal of Statistics, Series A (2008-)*, vol. 80, S1–S7, 2018.

[55] W. Samek, G. Montavon, A. Vedaldi, L. K. Hansen, and K.-R. Müller, *Explainable AI: interpreting, explaining and visualizing deep learning*. Springer Nature, 2019, vol. 11700.

[56] R. Caruana, Y. Lou, J. Gehrke, P. Koch, M. Sturm, and N. Elhadad, "Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission," in *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, 2015, pp. 1721–1730.

[57] S. Lapuschkin, S. Wäldchen, A. Binder, G. Montavon, W. Samek, and K.-R. Müller, "Unmasking clever hans predictors and assessing what machines really learn," *Nature communications*, vol. 10, no. 1, p. 1096, 2019.

[58] P. Linardatos, V. Papastefanopoulos, and S. Kotsiantis, "Explainable ai: A review of machine learning interpretability methods," *Entropy*, vol. 23, no. 1, p. 18, 2020.

[59] P.-J. Kindermans, S. Hooker, J. Adebayo, M. Alber, K. T. Schütt, S. Dähne, and D. Erhan, "The (un) reliability of saliency methods," *arXiv preprint arXiv:1711.00867*, 2017.

[60] D. Smilkov, N. Thorat, B. Kim, F. Viégas, and M. Wattenberg, "Smoothgrad: Removing noise by adding noise," 2017.

[61] M. Sundararajan, A. Taly, and Q. Yan, "Axiomatic attribution for deep networks," in *International Conference on Machine Learning*, PMLR, 2017, pp. 3319–3328.

[62] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *European conference on computer vision*, Springer, 2014, pp. 818–833.

[63] L. M. Zintgraf, T. S. Cohen, T. Adel, and M. Welling, "Visualizing deep neural network decisions: Prediction difference analysis," in *International Conference on Learning Representations*, 2017.

[64] R. C. Fong and A. Vedaldi, "Interpretable explanations of black boxes by meaningful perturbation," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 3429–3437.

[65] D. Erhan, Y. Bengio, A. Courville, and P. Vincent, "Visualizing higher-layer features of a deep network," *University of Montreal*, vol. 1341, no. 3, 2009.

[66] M. T. Ribeiro, S. Singh, and C. Guestrin, “"why should i trust you?": Explaining the predictions of any classifier,” in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 1135–1144.

[67] S. M. Lundberg and S.-I. Lee, “A unified approach to interpreting model predictions,” in *Advances in Neural Information Processing Systems*, 2017, pp. 4765–4774.

[68] M. T. Ribeiro, S. Singh, and C. Guestrin, “Anchors: High-precision model-agnostic explanations,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, 2018.

[69] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, and W. Samek, “On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation,” *PLOS ONE*, vol. 10, no. 7, e0130140, 2015.

[70] A. Shrikumar, P. Greenside, and T. Kundaje, “Learning important features through propagating activation differences,” in *Proceedings of the 34th International Conference on Machine Learning*, 2017, pp. 3145–3153.

[71] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, “Grad-cam: Visual explanations from deep networks via gradient-based localization,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 618–626.

[72] S. Abnar and W. Zuidema, “Quantifying attention flow in transformers,” in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020, pp. 2190–2202.

[73] I. T. Jolliffe, *Principal Component Analysis*. Springer, 2002.

[74] L. van der Maaten and G. Hinton, “Visualizing data using t-sne,” *Journal of Machine Learning Research*, vol. 9, pp. 2579–2605, 2008.

[75] L. McInnes, J. Healy, and J. Melville, “Umap: Uniform manifold approximation and projection for dimension reduction,” in *Advances in Neural Information Processing Systems*, 2018.

[76] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, “On calibration of modern neural networks,” in *International conference on machine learning*, PMLR, 2017, pp. 1321–1330.

[77] D. Hendrycks and T. Dietterich, "Benchmarking neural network robustness to common corruptions and perturbations," *Proceedings of the International Conference on Learning Representations*, 2019.

[78] B. Recht, R. Roelofs, L. Schmidt, and V. Shankar, "Do imagenet classifiers generalize to imagenet?" In *International conference on machine learning*, PMLR, 2019, pp. 5389–5400.

[79] P. W. Koh, S. Sagawa, H. Marklund, S. M. Xie, M. Zhang, A. Balsubramani, W. Hu, M. Yasunaga, R. L. Phillips, I. Gao, *et al.*, "Wilds: A benchmark of in-the-wild distribution shifts," in *International conference on machine learning*, PMLR, 2021, pp. 5637–5664.

[80] R. Geirhos, P. Rubisch, C. Michaelis, M. Bethge, F. A. Wichmann, and W. Brendel, "Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness," in *International conference on learning representations*, 2018.

[81] A. Athalye, N. Carlini, and D. Wagner, "Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples," in *International conference on machine learning*, PMLR, 2018, pp. 274–283.

[82] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "Deepfool: A simple and accurate method to fool deep neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2574–2582.

[83] F. Croce and M. Hein, "Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks," in *International conference on machine learning*, PMLR, 2020, pp. 2206–2216.

[84] T. B. Brown, D. Mané, A. Roy, M. Abadi, and J. Gilmer, "Adversarial patch," *arXiv preprint arXiv:1712.09665*, 2017.

[85] J. Ebrahimi, A. Rao, D. Lowd, and D. Dou, "Hotflip: White-box adversarial examples for text classification," *arXiv preprint arXiv:1712.06751*, 2017.

[86] D. Jin, Z. Jin, J. T. Zhou, and P. Szolovits, "Is bert really robust? natural language attack on text classification and entailment," *arXiv preprint arXiv:1907.11932*, 2019.

[87] A. Kantchelian, J. D. Tygar, and A. Joseph, "Evasion and hardening of tree ensemble classifiers," in *International conference on machine learning*, PMLR, 2016, pp. 2387–2396.

[88] K. Grosse, P. Manoharan, N. Papernot, M. Backes, and P. McDaniel, "On the (statistical) detection of adversarial examples," *arXiv preprint arXiv:1702.06280*, 2017.

[89] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," *arXiv preprint arXiv:1706.06083*, 2017.

[90] H. Zhang, Y. Yu, J. Jiao, E. Xing, L. El Ghaoui, and M. Jordan, "Theoretically principled trade-off between robustness and accuracy," in *International conference on machine learning*, PMLR, 2019, pp. 7472–7482.

[91] J. Cohen, E. Rosenfeld, and Z. Kolter, "Certified adversarial robustness via randomized smoothing," in *international conference on machine learning*, PMLR, 2019, pp. 1310–1320.

[92] D. Tsipras, S. Santurkar, L. Engstrom, A. Turner, and A. Madry, "Robustness may be at odds with accuracy," *arXiv preprint arXiv:1805.12152*, 2018.

[93] B. Settles, "Active learning literature survey," 2009.

[94] D. D. Lewis, "A sequential algorithm for training text classifiers: Corrigendum and additional data," in *Acm Sigir Forum*, ACM New York, NY, USA, vol. 29, 1995, pp. 13–19.

[95] S. Amershi, M. Cakmak, W. B. Knox, and T. Kulesza, "Power to the people: The role of humans in interactive machine learning," *AI magazine*, vol. 35, no. 4, pp. 105–120, 2014.

[96] S. Das, M. Ashrafuzzaman, F. T. Sheldon, and S. Shiva, "Network intrusion detection using natural language processing and ensemble machine learning," in *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*, IEEE, 2020, pp. 829–835.

[97] J. Li, H. Zhang, and Z. Wei, "The weighted word2vec paragraph vectors for anomaly detection over http traffic," *IEEE Access*, vol. 8, pp. 141 787–141 798, 2020.

[98] A. M. Vartouni, S. S. Kashi, and M. Teshnehlab, "An anomaly detection method to detect web attacks using stacked auto-encoder," in *2018 6th Iranian Joint Congress on Fuzzy and Intelligent Systems (CFIS)*, IEEE, 2018, pp. 131–134.

[99] C. Johnson, B. Khadka, R. B. Basnet, and T. Doleck, "Towards detecting and classifying malicious urls using deep learning.," *J. Wirel. Mob. Networks Ubiquitous Comput. Dependable Appl.*, vol. 11, no. 4, pp. 31–48, 2020.

[100] H. T. Nguyen, C. Torrano-Gimenez, G. Alvarez, S. Petrović, and K. Franke, "Application of the generic feature selection measure in detection of web attacks," in *Computational Intelligence in Security for Information Systems*, Springer, 2011, pp. 25–32.

[101] B. B. Gupta, K. Yadav, I. Razzak, K. Psannis, A. Castiglione, and X. Chang, "A novel approach for phishing urls detection using lexical based machine learning in a real-time environment," *Computer Communications*, vol. 175, pp. 47–57, 2021.

[102] J. Liu, X. Song, Y. Zhou, X. Peng, Y. Zhang, P. Liu, and D. Wu, "Deep anomaly detection in packet payload," *arXiv preprint arXiv:1912.02549*, 2019.

[103] C. Luo, Z. Tan, G. Min, J. Gan, W. Shi, and Z. Tian, "A novel web attack detection system for internet of things via ensemble classification," *IEEE Transactions on Industrial Informatics*, 2020.

[104] R. Flood, G. Engelen, D. Aspinall, and L. Desmet, "Bad design smells in benchmark nids datasets," in *2024 IEEE 9th European Symposium on Security and Privacy (EuroS&P)*, IEEE, 2024, pp. 658–675.

[105] M. Gniewkowski, H. Maciejewski, T. R. Surmacz, and W. Walentynowicz, "Http2vec: Embedding of http requests for detection of anomalous traffic," *arXiv preprint arXiv:2108.01763*, 2021.

[106] M. Gniewkowski, H. Maciejewski, T. Surmacz, and W. Walentynowicz, "Sec2vec: Anomaly detection in http traffic and malicious urls," in *Proceedings of the 38th ACM/SIGAPP Symposium on Applied Computing*, 2023, pp. 1154–1162.

[107] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," *arXiv preprint arXiv:1508.07909*, 2015.

[108] C. Wang, K. Cho, and J. Gu, "Neural machine translation with byte-level subwords," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, 2020, pp. 9154–9160.

[109] C. T. Giménez, A. P. Villegas, and G. Á. Marañón, "HTTP data set CSIC 2010," *Information Security Institute of CSIC (Spanish Research National Council)*, 2010.

[110] N. Moustafa and J. Slay, "Unsw-nb15: A comprehensive data set for network intrusion detection systems (unsw-nb15 network data set)," in *2015 military communications and information systems conference (MilCIS)*, IEEE, 2015, pp. 1–6.

[111] M. S. I. Mamun, M. A. Rathore, A. H. Lashkari, N. Stakhanova, and A. A. Ghorbani, "Detecting malicious urls using lexical analysis," in *International Conference on Network and System Security*, Springer, 2016, pp. 467–482.

[112] D. Zhang, F. Nan, X. Wei, S. Li, H. Zhu, K. McKeown, R. Nallapati, A. Arnold, and B. Xiang, "Supporting clustering with contrastive learning," *arXiv preprint arXiv:2103.12953*, 2021.

[113] X. Wu, L. Xiao, Y. Sun, J. Zhang, T. Ma, and L. He, "A survey of human-in-the-loop for machine learning," *Future Generation Computer Systems*, vol. 135, pp. 364–381, 2022.

[114] N. Ropiak, M. Gniewkowski, M. Swedrowski, M. Pogoda, K. Gawron, B. Bojanowski, and T. Walkowiak, "How to select samples for active learning? document clustering with active learning methodology," in *2023 27th International Conference on Engineering of Complex Computer Systems (ICECCS)*, 2023, pp. 42–50.

[115] L. Xuhong, Y. Grandvalet, and F. Davoine, "Explicit inductive bias for transfer learning with convolutional networks," in *International conference on machine learning*, PMLR, 2018, pp. 2825–2834.

[116] M. Gniewkowski and T. Walkowiak, "Assessment of document similarity visualisation methods," in *Language and Technology Conference*, Springer, 2019, pp. 348–363.

[117] C. Etmann, S. Lunz, P. Maass, and C.-B. Schönlieb, "On the connection between adversarial robustness and saliency map interpretability," *arXiv preprint arXiv:1905.04172*, 2019.

[118] A. Ghorbani, A. Abid, and J. Zou, "Interpretation of neural networks is fragile," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, 2019, pp. 3681–3688.

[119] M. Cai, X. Wang, F. Sohel, and H. Lei, "Contextual attribution maps-guided transferable adversarial attack for 3d object detection," *Remote Sensing*, vol. 16, no. 23, p. 4409, 2024.

[120] M. Gniewkowski, P. Walkowiak, P. Syga, M. Klonowski, and T. Walkowiak, "Do not trust me: Explainability against text classification," in *ECAI 2023*, IOS Press, 2023, pp. 875–882.

[121] M. Gniewkowski, P. Walkowiak, M. Klonowski, and T. Walkowiak, "Precise language deception: Xai driven targeted adversarial examples with restricted knowledge," in *Computational Science – ICCS 2025*, M. H. Lees, W. Cai, S. A. Cheong, Y. Su, D. Abramson, J. J. Dongarra, and P. M. A. Sloot, Eds., Cham: Springer Nature Switzerland, 2025, pp. 49–60.

[122] D. Jin, Z. Jin, J. T. Zhou, and P. Szolovits, "Is BERT really robust? natural language attack on text classification and entailment," *CoRR*, vol. abs/1907.11932, 2020.

[123] J. Li, S. Ji, T. Du, B. Li, and T. Wang, "TextBugger: Generating adversarial text against real-world applications," in *Proceedings 2019 Network and Distributed System Security Symposium*, Internet Society, 2019.

[124] R. Mroczkowski, P. Rybak, A. Wróblewska, and I. Gawlik, "HerBERT: Efficiently pretrained transformer-based language model for Polish," in *Proceedings of the 8th Workshop on Balto-Slavic Natural Language Processing*, Kiyv, Ukraine: Association for Computational Linguistics, Apr. 2021, pp. 1–10.

[125] A. Toral, R. Muñoz, and M. Monachini, "Named entity WordNet," in *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco: European Language Resources Association (ELRA), May 2008.

[126] K. Szyc, T. Walkowiak, and H. Maciejewski, "Checking robustness of representations learned by deep neural networks," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, Springer, 2021, pp. 399–414.

[127] K. Szyc, T. Walkowiak, and H. Maciejewski, "Beyond overall accuracy: Exploring per-class performance reveals robustness and safety gaps in deep models," in *2024 IEEE International Conference on Data Mining Workshops (ICDMW)*, IEEE, 2024, pp. 833–841.

[128] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A next-generation hyperparameter optimization framework," in *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, 2019, pp. 2623–2631.

[129] M. Andriushchenko, F. Croce, N. Flammarion, and M. Hein, "Square attack: A query-efficient black-box adversarial attack via random search," in *European conference on computer vision*, Springer, 2020, pp. 484–501.

[130] D. Hendrycks, N. Mu, E. D. Cubuk, B. Zoph, J. Gilmer, and B. Lakshminarayanan, "Augmix: A simple data processing method to improve robustness and uncertainty," *arXiv preprint arXiv:1912.02781*, 2019.

[131] T. Walkowiak, M. Klonowski, M. Gniewkowski, and P. Walkowiak, "Towards robust language models: Xai-driven generation and mitigation of targeted adversarial examples under restricted knowledge," *Available at SSRN 5828846*,

[132] G. Mak, M. Gniewkowski, P. Walkowiak, and A. Janz, "An empirical assessment of llm-based approaches to malicious webpage detection," in *Computational Science – ICCS 2025*, M. H. Lees, W. Cai, S. A. Cheong, Y. Su, D. Abramson, J. J. Dongarra, and P. M. A. Sloot, Eds., Cham: Springer Nature Switzerland, 2025, pp. 162–176.

[133] T. Walkowiak and M. Gniewkowski, "Visualisation of document similarities based on word embedding models for polish," *Wydawnictwo Nauka i Innowacje, Poznań*, pp. 148–151, 2019.

[134] T. Walkowiak and M. Gniewkowski, "Evaluation of vector embedding models in clustering of text documents," in *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*, 2019, pp. 1304–1311.

[135] T. Walkowiak and M. Gniewkowski, "Distance measures for clustering of documents in a topic space," in *International Conference on Dependability and Complex Systems*, Springer, 2019, pp. 544–552.

[136] M. Gniewkowski, "An overview of dos and ddos attack detection techniques," in *International Conference on Dependability and Complex Systems*, Springer, 2020, pp. 233–241.

[137] M. Marcińczuk, M. Gniewkowski, T. Walkowiak, and M. Będkowski, "Text document clustering: Wordnet vs. tf-idf vs. word embeddings," in *Proceedings of the 11th global wordnet conference*, 2021, pp. 207–214.

[138] M. Gniewkowski, H. Maciejewski, and T. Surmacz, "Anomaly detection techniques for different ddos attack types," in *International Conference on Dependability and Complex Systems*, Springer, 2022, pp. 63–78.

[139] W. Walentynowicz, M. Piasecki, and M. Gniewkowski, "Classification and generation of derivational morpho-semantic relations for polish language," in *International Conference on Computational Science*, Springer, 2022, pp. 244–251.

[140] N. Ropiak, M. Gniewkowski, M. Swedrowski, M. Pogoda, K. Gawron, B. Bojanowski, and T. Walkowiak, "How to select samples for active learning? document clustering with active learning methodology," in *2023 27th International Conference on Engineering of Complex Computer Systems (ICECCS)*, IEEE, 2023, pp. 42–50.

[141] T. Walkowiak, M. Gniewkowski, M. Pogoda, and N. Ropiak, "Anonymizer for polish language," *Wojciechowski A.(Ed.), Lipiński P.(Ed.)., Progress in Polish Artificial Intelligence Research 4, Seria: Monografie Politechniki Łódzkiej Nr. 2437, Wydawnictwo Politechniki Łódzkiej, Łódź 2023, ISBN 978-83-66741-92-8, doi: 10.34658/9788366741928.*, 2023.

[142] J. Kocoń, M. Piasecki, A. Janz, T. Ferdinan, Ł. Radliński, B. Koptyra, M. Oleksy, S. Woźniak, P. Walkowiak, K. Wojtasik, J. Moska, T. Naskręt, B. Walkowiak, M. Gniewkowski, K. Szyc, D. Motyka, D. Banach, J. Dalasiński, E. Rudnicka, B. Alberski, T. Walkowiak, A. Szczęsny, M. Markiewicz, T. Bernaś, H. Mazur, K. Żyta, M. Tykierko, G. Chodak, T. Kajdanowicz, P. Kazienko, A. Karlińska, K. Seweryn, A. Kołos, M. Chrabąszcz, K. Lorenc, A. Krasnodębska, A. Wilczek, K. Dziewulska, P. Betscher, Z. Cieślińska, K. Kowol, D. Mikoś, M. Trzciński, D. Krutul, M. Kozłowski, S. Dadas, R. Poświata, M. Perełkiewicz, M. Grębowiec, M. Kazuła, M. Białas, R. Roszko, D. Roszko, J. Vaičenonienė, A. Utka, P. Levchuk, P. Kowalski, I. Prawdzic-Jankowska, M. Ogrodniczuk, M. Borys, A. Bulińska, W. Gumienna, W. Kieraś, D. Komosińska, K. Krasnowska-Kieraś, Ł. Kobyliński, M. Lewandowska, M. Łaziński, M. Łątkowski, D. Mastalerz, B. Milewicz, A. A. Mykowiecka, A. Peljak-Łapińska, S. Penno, Z. Przybysz, M. Rudolf, P. Rybak, K. Saputa, A. Tomaszewska, A. Wawer, M. Woliński, J. Wołoszyn, A. Wróblewska, B. Żuk, F. Żarnecki, K. Kaczyński, A. Cichosz, Z. Deckert, M. Garnys, I. Grabarczyk, W. Janowski, S. Karasińska, A. Kujawiak, P. Misztela, M. Szymańska, K. Walkusz, I. Siek, J. Kwiatkowski, and P. Pęzik, *Pllum: A family of polish large language models*, 2025.

Appendix A – Personal achievements

List of scientific publications

1. T. Walkowiak, M. Klonowski, M. Gniewkowski, and P. Walkowiak, "Towards robust language models: Xai-driven generation and mitigation of targeted adversarial examples under restricted knowledge," *Available at SSRN 5828846*,

2. G. Mak, M. Gniewkowski, P. Walkowiak, and A. Janz, "An empirical assessment of llm-based approaches to malicious webpage detection," in *Computational Science – ICCS 2025*, M. H. Lees, W. Cai, S. A. Cheong, *et al.*, Eds., Cham: Springer Nature Switzerland, 2025, pp. 162–176

3. M. Gniewkowski, P. Walkowiak, M. Klonowski, and T. Walkowiak, "Precise language deception: Xai driven targeted adversarial examples with restricted knowledge," in *Computational Science – ICCS 2025*, M. H. Lees, W. Cai, S. A. Cheong, *et al.*, Eds., Cham: Springer Nature Switzerland, 2025, pp. 49–60

4. M. Gniewkowski and T. Walkowiak, "Assessment of document similarity visualisation methods," in *Language and Technology Conference*, Springer, 2019, pp. 348–363

5. T. Walkowiak and M. Gniewkowski, "Visualisation of document similarities based on word embedding models for polish," *Wydawnictwo Nauka i Innowacje, Poznań*, pp. 148–151, 2019

6. T. Walkowiak and M. Gniewkowski, "Evaluation of vector embedding models in clustering of text documents," in *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*, 2019, pp. 1304–1311

7. T. Walkowiak and M. Gniewkowski, “Distance measures for clustering of documents in a topic space,” in *International Conference on Dependability and Complex Systems*, Springer, 2019, pp. 544–552

8. M. Gniewkowski, “An overview of dos and ddos attack detection techniques,” in *International Conference on Dependability and Complex Systems*, Springer, 2020, pp. 233–241

9. M. Marcińczuk, M. Gniewkowski, T. Walkowiak, and M. Będkowski, “Text document clustering: Wordnet vs. tf-idf vs. word embeddings,” in *Proceedings of the 11th global wordnet conference*, 2021, pp. 207–214

10. M. Gniewkowski, H. Maciejewski, and T. Surmacz, “Anomaly detection techniques for different ddos attack types,” in *International Conference on Dependability and Complex Systems*, Springer, 2022, pp. 63–78

11. M. Gniewkowski, H. Maciejewski, T. R. Surmacz, and W. Walentynowicz, “Http2vec: Embedding of http requests for detection of anomalous traffic,” *arXiv preprint arXiv:2108.01763*, 2021

12. M. Gniewkowski, H. Maciejewski, T. Surmacz, and W. Walentynowicz, “Sec2vec: Anomaly detection in http traffic and malicious urls,” in *Proceedings of the 38th ACM/SIGAPP Symposium on Applied Computing*, 2023, pp. 1154–1162

13. W. Walentynowicz, M. Piasecki, and M. Gniewkowski, “Classification and generation of derivational morpho-semantic relations for polish language,” in *International Conference on Computational Science*, Springer, 2022, pp. 244–251

14. N. Ropiak, M. Gniewkowski, M. Swedrowski, *et al.*, “How to select samples for active learning? document clustering with active learning methodology,” in *2023 27th International Conference on Engineering of Complex Computer Systems (ICECCS)*, IEEE, 2023, pp. 42–50

15. T. Walkowiak, M. Gniewkowski, M. Pogoda, and N. Ropiak, “Anonymizer for polish language,” *Wojciechowski A.(Ed.), Lipiński P.(Ed.)., Progress in Polish Artificial Intelligence Research 4, Seria: Monografie Politechniki Łódzkiej Nr. 2437, Wydawnictwo Politechniki Łódzkiej, Łódź 2023, ISBN 978-83-66741-92-8, doi: 10.34658/9788366741928.*, 2023

16. M. Gniewkowski, P. Walkowiak, P. Syga, *et al.*, "Do not trust me: Explainability against text classification," in *ECAI 2023*, IOS Press, 2023, pp. 875–882

17. J. Kocoń, M. Piasecki, A. Janz, *et al.*, *Pllum: A family of polish large language models*, 2025