Adaptive DeSeTra: Adaptive Deformable-span Self-attention Transformer for LLM Security

Konstantinos Giapantzis, Panagiotis Bountakas, Apostolis Zarras, Aristeidis Farao, Vaios Bolgouras, Christos Xenakis

University of Piraeus, Greece

Abstract

Large Language Models (LLMs) remain vulnerable to prompt-based attacks such as jailbreaks and prompt injection, highlighting the need for security mechanisms that not only detect malicious intent but also determine whether an attack actually succeeds. In this paper, we introduce Adaptive DeSeTra, a novel security-centric Transformer model designed for deployment that mirrors the real-world attack pipeline through two layers: (*i*) *intent detection* via multi-class prompt classification into `Benign`, `Jailbreak` and `Prompt Injection`; and (*ii*) *impact assessment* via response-level classification into `Compliant` or `Refusal`. Prompt-level intent identification enables low-latency routing to appropriate guardrails, policies, and monitoring controls before a malicious prompt reaches the target LLM. Conversely, response-level evaluation quantifies whether the adversarial attempt resulted in compliance or refusal, providing an outcome-based measure of attack effectiveness that supports risk assessment and enables more informative security benchmarking than intent detection alone. Experiments demonstrate strong performance across both layers: 99.35% Accuracy/F1/Precision/Recall with 99.85% AUC for prompt classification, and 99.80% for the same metrics with 99.98% AUC for impact assessment, positioning Adaptive DeSeTra as a strong candidate for deployment-oriented LLM security monitoring and evaluation.

Keywords: LLM security, prompt classification, impact assessment, jailbreak, prompt injection

1. Introduction

Modern Artificial Intelligence (AI) development increasingly depends on versatile foundation capabilities enabled by Large Language Models (LLMs), allowing a single pretrained model to support a wide range of downstream functions [1]. In practice, LLMs combine strong natural-language understanding with fluent, context-aware generation, which has driven their adoption across domains including software engineering, robotics and cybersecurity [2]. More recently, agentic AI [3] has advanced this paradigm by integrating LLMs into applications as autonomous decision-making components [4]. In cybersecurity, this transition extends beyond static threat detection toward systems capable of planning, invoking security tools, and executing multi-step defensive workflows with limited human intervention [5]. However, to perform these functions, agentic systems must routinely access and reason over sensitive assets, including personal information, authorization credentials, infrastructure and security configurations, and proprietary code or operational procedures [6]. Consequently, the security of critical infrastructure increasingly hinges on the integrity of the underlying LLMs and their robustness against malicious influence and manipulation [7].

This reliance creates a strong incentive for adversaries to target LLMs by manipulating the textual instructions and contextual inputs that guide their decisions, motivating both jailbreak and prompt injection attacks aimed at misdirecting model behavior [8]. Jailbreak attacks use adversarial prompts to override built-in safety guardrails and induce outputs the model was designed to avoid, including misinformation, toxic or biased language, and illicit instructions [9]. Prompt injections achieve a similar effect by embedding deceptive instructions into the model's input to hijack its original directives, leading the system to ignore developer guidelines and, in some cases, leak sensitive information, confidential data, or perform unauthorized actions [10]. Significantly, jailbreak and prompt injection attacks are now recognized as a critical security risk for LLM applications, threatening user privacy, system reliability, and the overall safety of AI-powered applications [11].

To mitigate jailbreak and prompt-injection risks, existing solutions broadly follow a defense-in-depth pattern that combines detection with prevention [12]. Modern defenses are dominated by prompt-level detection framed as binary classification, typically built on curated datasets and implemented either as fine-tuned Transformers or lightweight embedding combined with Machine Learning (ML) pipelines [13]. Jailbreak work emphasizes curated prompt

datasets and Transformer-based or embedding-based detectors, increasingly accounting for multimodal attack channels [14]. Despite these advancements, recent evaluations indicate that sophisticated jailbreak and prompt injection methods can still attain high success rates, often exceeding 90%, against current defenses [15].

To address these challenges, we introduce Adaptive DeSeTra, a Transformer-based classifier that advances adversarial prompt evaluation along three novel aspects: (i) multi-class prompt classification, (ii) impact assessment, and (iii) an adaptive multi-head self-attention mechanism. The main novelty of Adaptive DeSeTra, is that it performs fine-grained, multiclass attribution, separating distinct prompt types (benign, jailbreaks, prompt injections) performing not only detection of adversarial prompts but also classification to the relevant categories driving context-aware mitigations. In addition, Adaptive DeSeTra uniquely provides impact assessment by analyzing the target LLM's response to determine whether the model complied or refused, thereby delivering an explicit, outcome-oriented signal of attack success or guardrail effectiveness that existing detectors typically omit. These capabilities are enabled by Adaptive DeSeTra 's novel architectural design, which combines a Transformer encoder with an adaptive multi-head self-attention mechanism for variable-range context aggregation and a convolutional augmentation module embedded in the first encoder layer. This design captures key local patterns alongside longer-range dependencies, yielding robust generalization without relying on large-scale linguistic pre-training.

In summary, we make the following main contributions:

- We introduce Adaptive DeSeTra, a Transformer model that combines adaptive multi-head self-attention spans with early convolutional augmentation, enabling robust multi-scale context modeling for LLM security.
- We improve adversarial prompt classification by adopting a multi-class formulation that accurately distinguishes benign, jailbreak and prompt-injection attacks, while also reliably identifying LLM compliance versus refusal in response to adversarial inputs.
- We curate a large-scale, heterogeneous security-focused training corpus from diverse sources through controlled synthesis and rigorous filtering, improving resilience to evolving attack strategies while mitigating benchmark-specific bias.

2. Related Work

2.1. Jailbreak Attacks Detection

Sayeedi et al. [14] utilized synthetic data generation to train JailBreakBERT and JailBreakRoBERTa, achieving F1 scores above 96.7%. While effective, the reliance on heavy augmentation raises concerns regarding data quality and real-world distribution shifts. Complementarily, Galinkin and Sablotny [16] aggregated three public sources (DAN [17], garak [18], jackhhao [19]) into 17,085 deduplicated samples (1,580 jailbreak; 15,505 benign), trained lightweight embedding-based classifiers tuned via 5-fold CV to optimize F1 under severe imbalance, and reported their best test model (Random Forest over Snowflake embeddings) at 83.33% F1 (FPR 0.0092, FNR 0.2188), motivating more sophisticated classifiers for improved detection. Finally, Zhang et al. [20] extended detection beyond text-only prompts with JailGuard, a mutation-and-divergence detector for prompt-based attacks across text and image modalities, evaluated on an 11,000-item dataset covering 15 attack types on GPT-3.5-Turbo-1106 (text) and MiniGPT-4 (images), achieving 86.14% Accuracy / 80.58% Precision / 86.10% Recall on text and 82.90% Accuracy / 74.00% Precision / 88.25% Recall on images.

2.2. Prompt Injection Attack Detection

Rahman et al. [13] achieved 96.55% accuracy on the DeepSet benchmark [21] using BERT embeddings and Logistic Regression, though the dataset was limited. Ayub and Majumdar [22] addressed scale by curating over 467,000 prompts, finding that Random Forest with OpenAI embeddings reached 86.8% F1, though this still trails top-tier detectors. Using DeepSet as a compact comparison point, Jain et al. [23] benchmarked representations (TF-IDF, Word2Vec, Bag-of-Words, and BERT embeddings from bert-base-uncased) and reported their best model as an RNN with BERT embeddings at 94.74% test accuracy. Kokkula et al. [24] proposed Palisade, a three-stage pipeline combining a heuristic filter, a BERT-based classifier, and a Groq-hosted LLM verifier (flagging if any layer triggers), trained on an English-filtered subset of the deepface/prompt-injections data (reported as 546 prompts; 80/20 split) and achieving 88.78% Accuracy, 78.26% Precision, 97.30% Recall, and 86.75% F1 from the reported confusion matrix, though without deep per-layer analysis. Closest in spirit to single-backbone supervised adaptation, Rahman et al. [25] studied XLM-RoBERTa on DeepSet (546 train / 116 test), contrasting zero-shot performance (55.17% Accuracy,

62.32% F1) with fine-tuning (up to 50 epochs), which boosted results to 99.14% Accuracy, 100% Precision, 98.33% Recall, and 99.16% F1. Finally, Deshpande et al. [26] explored two lightweight neural baselines under an 80/20 split, a two-layer LSTM and a feed-forward network, reporting 91.69% vs. 92.52% Accuracy, with modest F1 (69% for LSTM; 73% for feed-forward) driven by lower recall/precision trade-offs.

3. Adaptive DeSeTra

In this section, we present Adaptive DeSeTra, a Transformer-based model for LLM security that integrates two complementary stages: (*i*) prompt classification, which separates benign, jailbreak, and prompt-injection inputs, and (*ii*) impact assessment, which determines whether an adversarial attempt alters the target LLM's behavior. We also describe the architecture of the Adaptive DeSeTra encoder and its novel adaptive attention mechanism, alongside its convolutional augmentation module.

3.1. Overview

Existing detection frameworks uniformly treat adversarial prompt identification as binary classification, distinguishing only between malicious and benign inputs without differentiating among structurally distinct attack classes. This formulation conflates jailbreak prompts, which seek to elicit disallowed content despite policy constraints, with prompt-injection attempts, which override the instruction hierarchy to redirect model behavior through manipulated system or tool instructions. Because these attack types exploit different failure modes and require different mitigation strategies, binary classification limits the specificity of downstream security responses. Furthermore, the problem of impact assessment in LLM security, that is, determining from the target model's response whether an adversarial attempt actually resulted in compliance or was refused, remains unaddressed. Current approaches stop at intent detection and do not verify the behavioral outcome, leaving a gap between flagging a potentially harmful prompt and measuring whether it succeeded in altering model behavior.

Adaptive DeSeTra addresses both limitations through three contributions. First, it reformulates prompt-level detection as multi-class attribution, explicitly separating benign prompts, jailbreak attempts and prompt-injection attacks within a single classifier, thereby enabling class-specific routing, logging, and response policies that binary detectors cannot support.

Building on this finer-grained intent signal, it introduces a response-level impact assessment stage that classifies the target LLM's output as compliance or refusal, providing a direct, outcome-based measurement of attack success that links the identified threat type to its observed effect on model behavior. Both stages share the same architectural backbone, a deformable-span self-attention mechanism in which each attention head learns an independent context range, complemented by an early convolutional refinement stage that strengthens local feature extraction without large-scale linguistic pretraining, replacing separate detection components with a unified model family that couples intent detection with outcome verification end to end.

3.2. Prompt Classification

In the field of LLM security, the earliest and most consequential control point is the user's input, since it is the primary interface through which adversaries attempt to influence model behavior. Prompt classification is the process of automatically analyzing an input prompt and assigning it to predefined intent- and risk-oriented categories (Benign, Jailbreak, or Prompt Injection) so the system can take an immediate, policy-driven action. This is necessary because malicious prompts are engineered instructions that try to (*i*) coerce the model into violating safety constraints or producing harmful outputs (jailbreaks), or (*ii*) override instruction hierarchy and application rules to redirect the model's behavior regardless of the user's legitimate request (prompt injections). As a result, jailbreaks primarily compromise the session of the user issuing the attack prompt. In contrast, prompt injections can compromise the broader system behavior and, in tool-using applications, potentially impact other users and organizations.

To address this risk surface, Adaptive DeSeTra is used as a dedicated threat-detection classifier at the ingress of the LLM pipeline, where it evaluates each incoming prompt before forwarding it to the target model. By performing prompt classification in real time, Adaptive DeSeTra deterministically enforces: benign prompts proceed normally, while suspicious prompts can be blocked, routed to stricter guardrails, or subjected to additional verification. Beyond binary detection, Adaptive DeSeTra is designed to identify the type of malicious prompt (benign/jailbreak/prompt injection), which is operationally critical because these attack classes require different mitigations, logging semantics, and incident response actions. In practice, this turns prompt classification into an effective security control that both reduces exposure to adversarial prompting and improves visibility by producing

structured, high-signal labels for monitoring and governance.

3.3. Impact Assessment

On the other hand, impact assessment refers to the structured process of estimating the severity, scope and likelihood of harm that may arise when a malicious prompt succeeds, so that defenses and response actions can be prioritized according to risk. Unlike prompt classification, which is intent-oriented, impact assessment is outcome-oriented and operates solely on the generated response. Its purpose is not to infer whether the original prompt was adversarial or benign, but to determine whether the LLM ultimately complied with or refused the issued request. Determining the actual impact of an adversarial interaction is increasingly non-trivial because model behavior cannot be reliably inferred through simple, static inspection of responses. Different LLMs exhibit different refusal styles and compliance patterns, and even the exact model may vary its phrasing across prompts in ways that obscure whether it actually followed the instruction. In practice, a response may begin with a refusal statement and later provide the requested harmful content, or it may partially comply while masking key details behind disclaimers. This variability in response structure, refusal language, and latent compliance makes attack success harder to assess consistently across prompts and models, motivating an impact-assessment process based on the final behavioral outcome rather than the response's initial tone.

Given this ambiguity in refusal and compliance signals, Adaptive DeSeTra enables impact assessment by turning response analysis into consistent, risk-oriented measurements that can directly inform mitigation and response. Impact assessment can be applied in multiple scenarios and for different purposes, either as a complement to prompt classification or as a stand-alone evaluation capability. In a combined workflow, Adaptive DeSeTra can quantify how a target LLM responds to jailbreak and prompt-injection attempts by measuring comply and refuse outcomes at scale, thereby estimating success rates and identifying vulnerability trends that are otherwise obscured by differences in response phrasing. If the results indicate that the model is vulnerable and that comply outcomes outweigh refuse outcomes, Adaptive DeSeTra can then be deployed operationally as a prompt classification gate to prevent malicious prompts from reaching the target model. Independently, impact assessment can also be used as a dedicated tool for evaluating and improving resilience against jailbreak and prompt injection attacks, providing a more complete picture of LLM behavior by systematically characterizing

failure modes, partial compliance patterns, and the conditions under which safeguards break down.

3.4. Adaptive DeSeTra's Architecture

Adaptive DeSeTra is a Transformer-based architecture designed to provide a security-focused language-understanding backbone with high precision. The main objective is to preserve the practical advantages of standard Transformer models while introducing a principled notion of adaptivity in how context is processed. The closest prior work on adaptive attention is that of Sukhbaatar et al. [27], who introduced head-specific learnable spans in a sequential autoregressive Transformer for character-level language modeling. However, our proposed approach differs in both architecture and objective. Sukhbaatar et al. focus on causal next-token prediction, where adaptive span is used to control the amount of past context each head can access while improving efficiency over long sequences.

In contrast, Adaptive DeSeTra is an encoder-based model for classification, where attention operates over the full input sequence to support security-oriented decisions. Accordingly, rather than applying a soft span mask to causal attention weights, Adaptive DeSeTra uses a deformable head-wise span gate on the scaled attention scores before softmax, combined with padding-aware masking. In simple terms, Adaptive DeSeTra relies on multi-head self-attention where each attention head learns not only *what* tokens to attend to, but also *how far* it should look within the sequence, making the attention mechanism inherently adaptive. We refer to this learned head-specific range as the *attention span*. Instead of using a fixed, globally defined context window, each head adapts its span during training, yielding a set of specialized heads that may focus on local patterns, medium-range structure, or longer-range dependencies, according to what is most informative for the task.

Following the canonical Transformer design, Adaptive DeSeTra processes inputs through a token embedding layer over a vocabulary of 128,100 tokens into a stack of 4 encoder layers, each with hidden dimensionality 1536 and 8 attention heads. Absolute positional embeddings are used for sequences of up to 512 tokens. The embedding representations are regularized with dropout 0.1 and normalized before entering the encoder stack. Each encoder layer consists of an adaptive multi-head self-attention sublayer followed by a position-wise feed-forward sublayer. In the self-attention block, the query, key, value and output projections preserve the model dimensionality, while

attention probabilities are regularized with dropout 0.1. In the feed-forward block, the hidden representation is expanded from 1536 to 6144, transformed with a GELU activation function, and projected back to 1536. Residual connections and layer normalization with $\epsilon = 10^{-7}$ are applied around the attention and feed-forward transformations to stabilize optimization.

3.4.1. Adaptive-span multi-head self-attention mechanism

The core innovation is realized in the multi-head self-attention mechanism, which augments scaled dot-product attention with an adaptive, deformable span kernel. After standard linear projections are transformed into queries, keys and values, the attention logits are multiplicatively modulated by a continuous gate that is learned independently per attention head. Each head maintains a span parameter R_h, represented as a raw, learnable scalar, and mapped via a softplus transform to enforce positivity. Given directed relative index offsets $Z = i - j$ between query and key positions, the gate produces values in $[0, 1]$ that progressively attenuate contributions beyond the learned span, with smoothness controlled by a softness hyperparameter for the adaptive span transition ϵ_{span}, defaulted to 0.01. This formulation makes adaptivity explicit. Attention does not merely select relevant tokens based on content similarity; it also learns how far to look, and it can do so heterogeneously across heads within the same layer. The attention module is additionally engineered for robust inference, including correct reconciliation of padding constraints with key/value caching and safeguards that prevent numerical instability in fully masked rows prior to the softmax. Consequently, Adaptive DeSeTra retains a unified attention primitive that is ready for deployment in encoder settings.

As illustrated in Figure 1, the proposed attention mechanism receives query, key and value input embeddings $X^Q \in \mathbb{R}^{B\times L_q\times d}$, $X^K \in \mathbb{R}^{B\times L_k\times d}$ and $X^V \in \mathbb{R}^{B\times L_v\times d}$, where B is the batch size, d is the model dimension and H is the number of attention heads with per-head dimension $d_h = d/H$. The module begins with QKV linear projections and reshapes the results into multiple heads:

$$Q = \text{reshape}(X^Q W_Q + b_Q) \in \mathbb{R}^{B\times H\times L_q\times d_h}, \tag{1}$$

$$K = \text{reshape}(X^K W_K + b_K) \in \mathbb{R}^{B\times H\times L_k\times d_h}, \tag{2}$$

$$V = \text{reshape}(X^V W_V + b_V) \in \mathbb{R}^{B\times H\times L_v\times d_h}, \tag{3}$$

with parameters $W_Q, W_K, W_V \in \mathbb{R}^{d\times d}$ and $b_Q, b_K, b_V \in \mathbb{R}^d$. The attention core is engineered to support key/value caching. When a past key–value pair

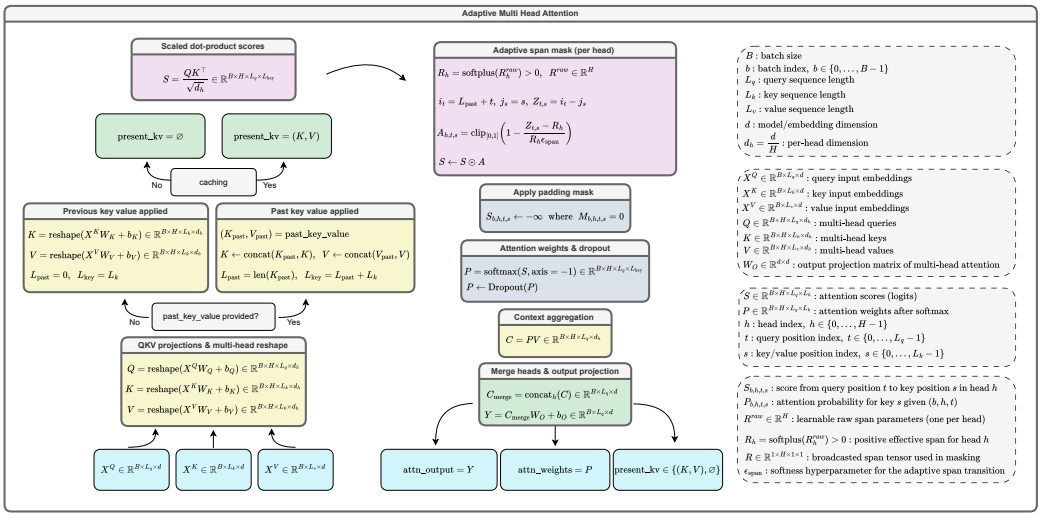


Figure 1: Adaptive DeSeTra's multi-head self-attention mechanism.

is provided, then the effective keys and values are formed by concatenation along the sequence dimension and the resulting cache length and effective key length are $L_{\text{past}} = \text{len}(K_{\text{past}})$, $L_{\text{key}} = L_{\text{past}} + L_k$. The scaled dot-product logits are computed as:

$$S = \frac{QK^\top}{\sqrt{d_h}} \in \mathbb{R}^{B\times H\times L_q\times L_{\text{key}}}. \tag{4}$$

The proposed method introduces an adaptive, head-wise span gate that multiplicatively modulates these logits with a continuous mask, thereby transforming attention range from a fixed design choice into a differentiable, learned parameter. Each head h maintains a raw span scalar R_h^{raw}, mapped to a strictly positive effective span using a softplus transform:

$$R_h = \text{softplus}(R_h^{\text{raw}}) > 0, \qquad R^{\text{raw}} \in \mathbb{R}^H. \tag{5}$$

To align span gating with caching, the mechanism constructs relative offsets using absolute indices. For each query position $t \in \{0, \ldots, L_q - 1\}$ define: $i_t = L_{\text{past}} + t$, and for each key position $s \in \{0, \ldots, L_{\text{key}} - 1\}$ define $j_s = s$, so the directed relative offset is $Z_{t,s} = i_t - j_s$. The adaptive span mask is then computed per head as a clipped ramp with softness parameter ϵ_{span}:

$$A_{h,t,s} = \text{clip}_{[0,1]}\left(1 - \frac{Z_{t,s} - R_h}{R_h \epsilon_{\text{span}}}\right), \tag{6}$$

and logits are gated as $S \leftarrow S \odot A$. Due to clipping, the gate has three regimes: $A_{h,t,s} = 1$ when $Z_{t,s} \leq R_h$, linear decay for $R_h < Z_{t,s} < R_h(1 + \epsilon_{\text{span}})$, and 0 once $Z_{t,s} \geq R_h(1 + \epsilon_{\text{span}})$.

This construction yields several advantages over standard attention and hard-window variants. First, it separates content-based compatibility from range-based eligibility. Classic attention relies exclusively on dot-product logits $S_{b,h,t,s}$ to suppress irrelevant tokens, thereby forcing distance preferences to be expressed implicitly through content similarity. In contrast, here distance preference is expressed explicitly via $A_{h,t,s}$, allowing each head to learn *"how far should I look"* independently of *"which tokens are compatible"*. Second, the gate is continuous and differentiable, unlike hard truncation. A hard window introduces discontinuities and brittle behavior, where tokens just outside the window become entirely invisible, and the window size requires manual tuning, whereas $A_{h,t,s}$ implements a deformable kernel with a soft boundary controlled by ϵ_{span}. For fixed R_h, the gate behaves like a saturating ramp, with a transition width that scales with $R_h \epsilon_{\text{span}}$. Third, because R_h is learned per head, the mechanism induces heterogeneous receptive fields within the same layer. Some heads can converge to a small R_h to model local dependencies, while others can maintain a larger R_h to capture longer-range structure, introducing an interpretable axis of diversity (range diversity) within multi-head attention.

Finally, in the multiplicative variant, the span prior is applied at the logit level by modifying the attention scores before softmax, thereby directly reshaping the relative score differences that determine the normalized attention distribution. Since softmax is sensitive to relative logit magnitudes, shaping logits before normalization changes which keys compete for probability mass. If a padding mask is provided, the module applies it by forcing invalid logits to $-\infty$. To prevent NaNs when an entire row is masked, the implementation stabilizes fully-masked rows by setting them to zero before softmax. Attention weights are then computed and regularized:

$$P = \text{softmax}(S, \text{axis} = -1) \in \mathbb{R}^{B \times H \times L_q \times L_{\text{key}}}, \tag{7}$$

$$P \leftarrow \text{Dropout}(P), \tag{8}$$

$$C = PV \in \mathbb{R}^{B \times H \times L_q \times d_h}. \tag{9}$$

Attention heads are then merged, and a learned output projection maps

the result back to the model dimension:

$$C_{\text{merge}} = \text{concat}_h(C) \in \mathbb{R}^{B \times L_q \times d}, \tag{10}$$

$$Y = C_{\text{merge}} W_O + b_O \in \mathbb{R}^{B \times L_q \times d}, \tag{11}$$

with $W_O \in \mathbb{R}^{d \times d}$ and $b_O \in \mathbb{R}^d$. The module returns Y as the attention output, P as the attention weights and K, V when caching is enabled. Overall, the proposed attention mechanism preserves the standard scaled dot-product attention pipeline while adding a head-wise, continuous adaptive span mask $A_{h,t,s}$ parameterized by $R_h = \text{softplus}(R_h^{\text{raw}})$, yielding explicit and learnable control over attention range, improved robustness in long-context regimes, and interpretable head specialization without sacrificing compatibility with padding constraints and key/value caching.

3.4.2. Encoder's Convolutional Augmentation Module

Beyond the adaptive attention mechanism, Adaptive DeSeTra introduces an encoder-level refinement through a convolutional augmentation module inserted immediately after the first encoder layer. This design is motivated by XLCNN [28] Transformer, which complements self-attention with convolutional processing in the feed-forward block. The purpose of this module is to strengthen the model's ability to capture short-range compositional and neighborhood-level patterns that may not be fully emphasized by self-attention alone, particularly in the early stages of learning. Placing it immediately after the first encoder layer is intentional, as this stage still retains relatively low-level token interactions and thus provides a suitable point for enriching the representations with local contextual cues before deeper layers progressively model longer-range dependencies through the deformable-span self-attention mechanism.

Concretely, the module applies three sequential 1D convolutions over the hidden representations, with each convolution preserving the channel dimensionality by mapping 1536 input channels to 1536 output channels. In the current implementation, all three convolutions use kernel size 3, single-group convolution, and padding that preserves the sequence length. After each convolution, padded positions are explicitly zeroed so that padding tokens do not contaminate the extracted local features. Dropout with rate 0.1 is then applied, followed by a `tanh` activation function. The output of the convolutional module is subsequently combined with the first encoder-layer representation through a residual connection and layer normalization with

$\epsilon = 10^{-7}$. Finally, the fused representation is processed by a compact two-layer linear mixing stage, in which both linear layers preserve the hidden dimensionality. After the first linear layer, dropout with rate 0.1 and `tanh` activation are again applied before the final linear transformation.

4. Evaluation

To provide a coherent and end-to-end evaluation of Adaptive DeSeTra as a security-oriented component, the remainder of this section is organized into two sequential stages that reflect the full attack pipeline. First, we study intent detection by training and benchmarking Adaptive DeSeTra for multi-class prompt classification (`Benign`, `Jailbreak`, `Prompt Injection`) and reporting dataset construction, training dynamics, validation behavior, and test-set performance, together with comparisons against both internal Transformer baselines and representative state-of-the-art results. Second, we move to impact assessment by evaluating whether malicious inputs actually elicit harmful behavior from the target LLM, formulated as response-level classification into `Compliant` or `Refusal`. For this stage, we repeat the same rigorous procedure (dataset description, training, validation, testing and comparative analysis) to quantify robustness in terms of observed outcomes rather than prompt intent alone.

To compare the performance and efficiency of Adaptive DeSeTra under a fair, balanced and reproducible baseline suite, we selected ALBERT, BERT, DeBERTa-v2, DistilBERT, ELECTRA, GPT-2, Longformer, RoBERTa, XL-CNN and XLNet as representative Transformer architectures spanning a broad range of model capacities. To ensure fairness in both prompt classification and impact assessment, all baseline models were trained under the same experimental protocol as Adaptive DeSeTra. Specifically, for each task, every baseline was trained on the same training split, validated on the same validation split and evaluated on the same held-out test split used by Adaptive DeSeTra. Moreover, the training, validation and test sets were strictly isolated and mutually exclusive, with no overlapping samples across splits, so that the reported performance metrics were not affected by data leakage or duplicate instances. In addition, all baseline models were initialized with the authors' publicly released default pre-trained weights, thereby benefiting from prior linguistic knowledge. In contrast, Adaptive DeSeTra was not pre-trained on any external dataset and was trained only on the task-specific datasets presented in this section. This design makes the comparison more

Dataset	Model	Benign	Jailbreak	Prompt Inj.	Total Prompts
LLM-LAT/benign-dataset	GPT-4	165,298	0	0	165,298
efgmarquez/jailbreak_dataset	–	0	27,090	0	27,090
jackhhao/jailbreak-classification	GPT-4, AIPRM, FlowGPT, awesome_chatgpt_prompts	640	666	0	1,306
rubend18/ChatGPT-Jailbreak-Prompts	GPT-4	0	79	0	79
reshabhs/SPML_Chatbot_Prompt_Injection	GPT-3.5, GPT-4, Llama-7b, Llama-13b	3,470	0	12,542	16,012
imoxto/prompt_injection_cleaned_dataset-v2	GPT-3.5-turbo, text-davinci-003	403,689	0	131,416	535,105
JasperLS/prompt-injections	–	399	0	263	662
GPT-4o-mini	GPT-4o-mini	0	28,758	0	28,758
DeepSeek	DeepSeek-V3.2	0	28,759	0	28,759
Proposed	All models above	68,395	68,548	68,843	205,786

Table 1: Proposed dataset for training the Adaptive DeSeTra model to distinguish between Benign prompts, Jailbreak attempts and Prompt Injection attacks

demanding for the proposed model and allows us to assess whether Adaptive DeSeTra, despite not inheriting external linguistic knowledge, can still outperform well-established Transformer architectures under identical data and evaluation conditions.

All experiments reported in this section were executed on a Linux server equipped with an AMD EPYC 7742 (x86_64) CPU and an NVIDIA A100-SXM4 GPU with 40 GB of VRAM, running Ubuntu 24.04-LTS. The processor provides 64 cores with 128 threads and operates at 2.25 GHz. The system also includes 125 GB of eight-channel DDR4 memory (3200 MT/s, 204.8 GB/s peak bandwidth) and its cache hierarchy consists of 96 KB L1 and 512 KB L2 per core, with a 256 MB shared L3 cache.

4.1. Prompt Classification

We first examine the prompt-classification stage, which assesses whether Adaptive DeSeTra can reliably distinguish benign prompts from jailbreak and prompt-injection attacks before they are forwarded to the target LLM. This stage forms the initial security checkpoint of the proposed pipeline, since accurate identification of malicious intent and attack type directly influences the effectiveness of subsequent protection mechanisms.

4.1.1. Proposed Dataset for malicious prompt detection

Accordingly, the data-collection process for training Adaptive DeSeTra is a key design requirement. The training set was constructed by integrating seven complementary, publicly available prompt corpora, each chosen to capture a distinct aspect of benign usage, jailbreak behavior, or prompt-injection

Class	Train	Validation	Testing
`Benign`	54,716	6,839	6,840
`Jailbreak`	54,869	6,839	6,840
`Prompt Injection`	55,164	6,839	6,840
`Total Prompts`	164,749	20,517	20,520

Table 2: Dataset splitting into training, validation, and testing subsets for prompt classification

attempts. To further broaden coverage of jailbreak variants, additional jailbreak prompts were synthesized using GPT-4o-mini and DeepSeek-V3.2. Table 1 summarizes the data used to train Adaptive DeSeTra for malicious input detection. The proposed dataset is divided into three classes: (*i*) a Benign class comprising prompts that do not pose a security risk to an LLM, (*ii*) a Jailbreak class including prompts intended to elicit policy-violating outputs from the targeted LLM, and (*iii*) a Prompt Injection class consisting of inputs crafted to override system-level instructions through malicious directives. To keep the distribution balanced between benign and malicious content, the proposed dataset consists of 68,395 Benign examples, 68,548 Jailbreak prompts, and 68,843 Prompt Injection commands, yielding a total of 205,786 records.

4.1.2. Preprocessing of the Prompt Classification Dataset

Before balancing and partitioning, the collected prompts underwent a deduplication and redundancy-control stage to reduce repeated prompt templates and minimize data leakage risk. This preprocessing was particularly important for the `Jailbreak` class, where the `efgmarquez/jailbreak_dataset` exhibited substantial duplication; from its initial 27,090 prompts, only 38% were retained, resulting in 10,286 unique records. In addition to source-level cleaning, the final corpus was further examined through similarity filtering. Each prompt was represented using TF–IDF vectors with unigram and bigram features, and intra-class cosine similarity was estimated with respect to the centroid of each class. The resulting mean similarity scores were 0.11 for `Benign`, 0.16 for `Jailbreak`, and 0.13 for `Prompt Injection`, indicating that the dataset is not dominated by highly repetitive formulations. Instead, the retained prompts preserve substantial lexical and structural diversity, which improves the robustness of the learned decision boundaries and supports a more reliable evaluation on unseen inputs.

After this cleaning and redundancy-reduction stage, the resulting cor-

pus was partitioned to support an objective and reproducible evaluation. Specifically, the dataset was divided into three mutually exclusive subsets: *i*) training, *ii*) validation and *iii*) testing, as reported in Table 2. The split was performed at the prompt level, so that each prompt was assigned to exactly one subset, with no shared, duplicated, or overlapping instances across training, validation, and testing. This strict separation was necessary to prevent data leakage and to ensure that performance on the validation and test sets reflected genuine generalization to unseen prompts rather than memorization of previously observed samples. The training subset was used to learn the decision boundaries among the three classes, while the validation subset supported hyperparameter tuning and model selection by monitoring generalization during training. The testing subset was kept fully isolated and used only for the final performance assessment. In addition, the split preserves a near-uniform class distribution across all subsets, reducing evaluation bias and ensuring that performance differences reflect the model's actual classification capability rather than class imbalance.

4.1.3. Training process

All baseline models including Adaptive DeSeTra were trained for 10 epochs using the prompt-classification splits reported in Table 2, which were constructed to be class-balanced across phases. Specifically, the training partition contains 54,716 *Benign*, 54,869 *Jailbreak* and 55,164 *Prompt Injection* prompts, while both validation and testing include 6,839 samples per class (20,517 total) and 6,840 samples per class (20,520 total), respectively. This balanced design was selected to prevent evaluation bias and to ensure that the learned decision boundaries are not driven by skewed class priors. The 10-epoch schedule provides sufficient exposure to diverse prompt patterns while limiting overfitting and unnecessary compute, consistent with the system's low-overhead goals. To further control class-dependent gradient bias during optimization, each epoch employed balanced stochastic sampling by randomly selecting

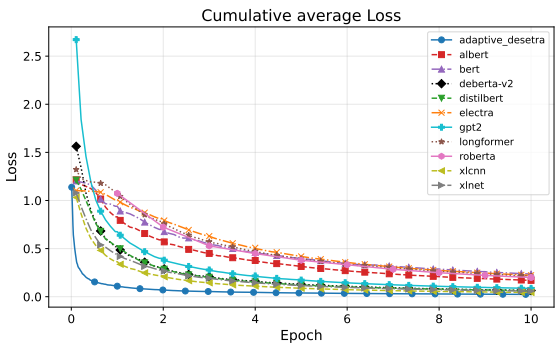


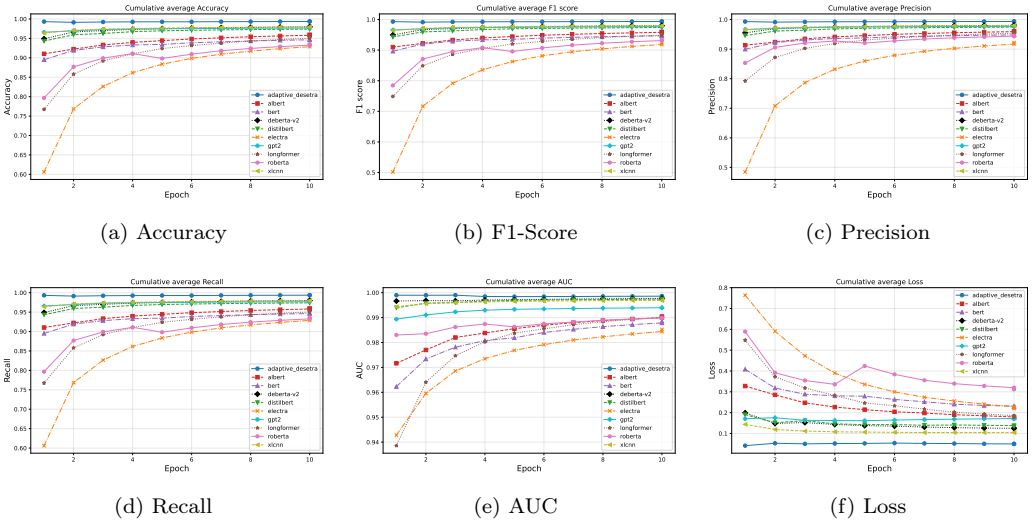
Figure 2: Cumulative average training loss across Transformer models for prompt classification

6,000 unique instances per class from the training subset, enforcing equal contribution of each category to parameter updates and improving training stability. Performance was monitored after every epoch on the validation set to track generalization and guide checkpoint selection, while the held-out test set was reserved for a single final assessment to avoid optimistic bias. Finally, the training, validation and testing partitions were strictly disjoint, ensuring no overlap and preventing data leakage across phases.

To determine the best performing architecture for malicious prompt detection, we conducted a comparative evaluation across multiple Transformer models and tracked their cumulative average loss throughout training. As illustrated in Figure 2, all models exhibit a monotonic decrease in cumulative loss, with the largest improvements occurring in the early epochs and progressively smaller gains thereafter. However, the convergence speed and terminal loss differ substantially across architectures. Adaptive DeSeTra converges the fastest and maintains the lowest cumulative loss throughout the training window, indicating the most efficient and stable optimization among the compared models. Specifically, its cumulative mean loss decreases from 0.1096 at the end of epoch 1 to 0.0699 at epoch 2 and 0.0541 at epoch 3, reaching 0.0241 by epoch 10, which corresponds to an approximate 78% reduction. Among the baselines, XLCNN and XLNet form the next-best convergence tier, ending with cumulative average losses of 0.0443 and 0.0584, respectively, followed closely by DistilBERT and DeBERTa-v2 at 0.0634 and 0.0647. GPT-2, despite starting with the highest initial loss, improves sharply during the first epoch and ultimately converges ahead of ALBERT, RoBERTa, Longformer, ELECTRA and BERT, which plateau at higher cumulative loss values. Given the same training objective and dataset, lower converged loss is generally associated with better classification performance. On this basis, the models are ordered from strongest to weakest as Adaptive DeSeTra, XLCNN, XLNet, DistilBERT, DeBERTa-v2, GPT-2, ALBERT, RoBERTa, Longformer, ELECTRA and BERT.

4.1.4. Validation results

During validation, each Transformer candidate was evaluated at the end of each training epoch on the held-out validation subset using the same metric suite: Accuracy, F1-score, Precision, Recall, AUC, and Loss. Figure 3 reports these quantities as cumulative averages over epochs, meaning that the value shown at epoch k is the running mean of the validation scores obtained from epoch 1 through epoch k. This design intentionally smooths epoch-



(a) Accuracy (b) F1-Score (c) Precision

(d) Recall (e) AUC (f) Loss

Figure 3: Validation curves reported as cumulative averages across epochs for different Transformer models on prompt classification.

level variability and emphasizes the stability of generalization, rather than isolated peak performance at a single epoch, which helps select a model that is consistently strong throughout training.

Across the validation results, using the epoch at which each model attains its highest validation accuracy, Adaptive DeSeTra delivers the strongest overall performance, achieving 99.52% Accuracy, F1-score, Precision and Recall, together with a 99.89% AUC and a Loss of 0.0406. Among the baselines, DeBERTa-v2 is the closest competitor, reaching 98.49% Accuracy and F1-score, 98.50% Precision, 98.49% Recall, 99.83% AUC, and a Loss of 0.1085. A second high-performing group includes XLNet (98.44% Accuracy, 99.71% AUC, 0.0922 Loss), XLCNN (98.28% Accuracy, 99.73% AUC, 0.1033 Loss), DistilBERT (98.16% Accuracy, 99.81% AUC, 0.0983 Loss) and Longformer (98.15% Accuracy, 99.69% AUC, 0.1005 Loss), all of which show strong validation behavior but remain below Adaptive DeSeTra. GPT-2 follows slightly behind with 97.93% Accuracy, 99.48% AUC and 0.1378 Loss, while ELECTRA (97.50% Accuracy, 99.29% AUC, 0.1174 Loss) and ALBERT (97.43% Accuracy, 99.51% AUC, 0.1274 Loss) form the next tier. By contrast, RoBERTa (97.09% Accuracy, 99.26% AUC, 0.1857 Loss) and BERT (96.07% Accuracy, 99.47% AUC, 0.1855 Loss) yield the weakest validation results among the evaluated baselines.

(a) Adaptive DeSeTra test-set confusion statistics for prompt classification.

Class	TP	FP	TN	FN	FPR	FNR
Benign	6,802	83	13,597	38	0.00607	0.00556
Jailbreak	6,771	39	13,641	69	0.00285	0.01009
Prompt Injection	6,814	11	13,669	26	0.00080	0.00380
Overall	20,387	133	40,907	133	0.00324	0.00648

(b) Adaptive DeSeTra test-set performance for prompt classification.

Class	Acc.	F1	Prec.	Recall
Benign	99.41	99.12	98.79	99.44
Jailbreak	99.47	99.21	99.47	98.99
Prompt Injection	99.82	99.73	99.84	99.62
Overall	99.35	99.35	99.35	99.35

Table 3: Adaptive DeSeTra results on the prompt classification test set, summarizing confusion matrix and per-class performance metrics

4.1.5. Testing results

Following model selection based on the validation process, the final evaluation was performed on the test subset, which remained strictly unseen throughout training and validation. In contrast to validation, which is used iteratively to monitor convergence and guide checkpoint selection, the test set provides a single, unbiased estimate of out-of-sample performance. It therefore serves as the definitive benchmark for assessing the prompt-classification capability of the selected model. The same metric suite was computed on the test partition, namely Accuracy, F1-score, Precision, Recall, AUC and Loss, to ensure comparability with the validation analysis and to quantify the expected performance under deployment conditions. Consistency between validation and testing outcomes indicates that the learned decision boundaries generalize beyond the tuning loop, while any degradation on the test set would signal overfitting or sensitivity to distributional shifts. Overall, this final evaluation confirms that Adaptive DeSeTra sustains its advantage under a fully held-out assessment and establishes its reliability for downstream security decisions.

Table 3a provides a detailed test-set assessment of Adaptive DeSeTra for prompt classification by reporting class-wise confusion statistics together with the corresponding false positive and false negative rates. Overall, the confusion counts are very small relative to the number of evaluated instances, and the derived False Positive Rate (FPR) and False Negative Rate (FNR) values remain consistently low across all classes, indicating both strong robustness and a favorable trade-off between missed detections and false alarms. Prompt Injection exhibits the cleanest separation, with negligible false positives and a very low false-negative rate, indicating that the model detects injection-style manipulations with high confidence. Jailbreak remains highly

accurate, with only a modest increase in false negatives, suggesting that a small fraction of jailbreak prompts are phrased in ways that more closely resemble injection patterns. Benign prompts exhibit a slightly higher false-positive rate than the other classes, consistent with conservative security screening settings where minimizing undetected attacks is often prioritized over occasional over-flagging.

Alongside the confusion statistics, standard classification metrics are reported per class, as shown in Table 3b, confirming uniformly high performance across the Benign, Jailbreak, and Prompt Injection categories. Prompt Injection again achieves the highest per-class scores, while Benign and Jailbreak remain comparably strong with only marginal differences between precision and recall. The "Overall" row reports identical values across Accuracy, F1-score, Precision, and Recall because the test set is class-balanced, meaning each class contributes equally to the global averages. Under this balanced regime, the macro-averaged metrics coincide, yielding a single aggregate value of 99.35%, reflecting the model's consistent behavior across all three classes.

4.1.6. Comparative Analysis

To ensure an objective and fair comparative evaluation, all Transformer models were trained under the same experimental procedure and assessed using an identical metric suite, including Accuracy, F1-score, Precision, Recall, AUC and Loss. In this section, the comparison focuses on the held-out test set, which provides the final, unbiased benchmark for evaluating and ranking generalization performance across architectures. Table 4a summarizes the comparative test-set performance of all Transformer baselines against the proposed Adaptive DeSeTra under an identical evaluation protocol. Across every metric, Adaptive DeSeTra achieves the strongest overall results, reaching 99.35% for Accuracy, F1-score, Precision and Recall, together with the highest AUC of 99.85%, indicating both highly reliable class decisions and separability between the three prompt categories. Among the baselines, XLNet emerges as the strongest overall competitor, achieving 98.40% on Accuracy, F1-score, Precision and Recall, together with a 99.68% AUC. It is followed by DeBERTa-v2, DistilBERT, Longformer and XLCNN, all of which exceed 98% on the primary metrics and maintain strong AUC values above 99.6%. GPT-2, ALBERT and ELECTRA form the next performance tier in the 97%–98% range, while RoBERTa and especially BERT yield the weakest accuracy-oriented results, despite BERT still reporting a high AUC of

(a) Test-set performance comparison across Transformer models on the proposed dataset

Model	Acc. (%)	F1 (%)	Prec. (%)	Rec. (%)	AUC (%)
Albert	97.41	97.41	97.42	97.41	97.46
Bert	96.04	96.04	96.07	96.04	99.51
Deberta-v2	98.26	98.26	98.27	98.26	99.76
Distilbert	98.16	98.16	98.18	98.16	99.81
Electra	97.22	97.22	97.26	97.22	99.36
GPT-2	97.65	97.65	97.69	97.65	99.37
Longformer	98.04	98.04	98.04	98.04	99.63
RoBERTa	96.46	96.46	96.46	96.46	98.95
XLCNN	98.03	98.03	98.05	98.03	99.65
XLNet	98.40	98.40	98.41	98.40	99.68
Adaptive DeSeTra	**99.35**	**99.35**	**99.35**	**99.35**	**99.85**

(b) Average system resource usage during model inference for prompt classification

Model	Params (Mil.)	CPU (%)	RAM (MB)	VRAM (MB)	Pred. Time (s)
Albert	13	1.732	43	578	0.0138
Bert	109	2.770	0	980	0.0288
Deberta-v2	313	2.051	4	1752	0.0186
Distilbert	67	1.925	0	800	0.0161
Electra	14	3.624	0	582	0.0271
GPT-2	124	3.536	32	1054	0.0264
Longformer	359	4.423	453	1998	0.0399
RoBERTa	355	1.523	405	1914	0.0158
XLCNN	331	2.061	128	1825	0.0148
XLNet	342	5.591	422	1885	0.0233
Adaptive DeSeTra	323	1.855	371	1794	0.0137

Table 4: Prompt classification comparison across Transformer models, reporting test-set performance and average inference-time resource usage

99.51%, which suggests that it preserves good class separability but produces less effective decision boundaries for this task.

Table 4b complements the accuracy-oriented comparison by reporting the average system resource footprint observed during the test phase, thereby highlighting the practical deployment cost of each model. These measurements were obtained on the NVIDIA A100-SXM4 GPU with batch size 32 and the reported prediction times include tokenization overhead. The results show that lightweight encoders such as ALBERT, DistilBERT and ELECTRA require the smallest parameter counts and correspondingly low VRAM usage, while larger backbones such as Longformer, RoBERTa, XLCNN, XLNet and DeBERTa-v2 demand substantially more accelerator memory. Longformer is the most resource-intensive configuration, with 359 million parameters, 453 MB RAM, 1,998 MB VRAM, and the slowest prediction time at 0.0399 sec. ELECTRA, despite its compact size of only 14 million parameters and its low RAM and VRAM footprint, exhibits relatively high inference latency at 0.0271 sec per prediction, second only to BERT at 0.0288 sec. In contrast, Adaptive DeSeTra, with 323 million parameters, 371 MB RAM, and 1,794 MB VRAM, achieves the fastest inference time at 0.0137 sec per prediction, slightly outperforming ALBERT at 0.0138 sec and XLCNN at 0.0148 sec. Its CPU utilization (1.855%) also remains low and close to the more lightweight baselines, indicating that the proposed design delivers its accuracy gains without a proportional increase in per-request inference cost.

To quantify the reliability of these comparisons, we additionally report 95% Wilson score [29] confidence intervals for each model's prompt classification Accuracy, Precision and Recall, thereby capturing the uncertainty

Model	Acc. 95% Wilson CI	Prec. 95% Wilson CI	Rec. 95% Wilson CI
Albert	[97.18, 97.62]	[97.19, 97.63]	[97.18, 97.62]
Bert	[95.76, 96.30]	[95.79, 96.33]	[95.76, 96.30]
Deberta-v2	[98.07, 98.43]	[98.08, 98.44]	[98.07, 98.43]
Distilbert	[97.97, 98.34]	[97.99, 98.36]	[97.97, 98.34]
Electra	[96.99, 97.44]	[97.03, 97.47]	[96.99, 97.44]
GPT-2	[97.43, 97.85]	[97.47, 97.88]	[97.43, 97.85]
Longformer	[97.84, 98.22]	[97.84, 98.22]	[97.84, 98.22]
RoBERTa	[96.20, 96.70]	[96.20, 96.70]	[96.20, 96.70]
XLCNN	[97.83, 98.21]	[97.83, 98.21]	[97.83, 98.21]
XLNet	[98.22, 98.56]	[98.22, 98.56]	[98.22, 98.56]
Adaptive DeSeTra	**[99.23, 99.45]**	**[99.23, 99.45]**	**[99.23, 99.45]**

Table 5: 95% Wilson score confidence intervals for overall Accuracy, Precision and Recall on prompt classification.

associated with finite test-set size. As shown in Table 5, Adaptive DeSeTra remains clearly separated from the strongest baseline, with non-overlapping confidence intervals. Overall, this separation indicates that the advantage of Adaptive DeSeTra over the strongest baseline is statistically robust, supporting its selection as the most effective configuration for prompt classification.

We further contrast our results with external evaluations reported in prior research, including studies that assess different models and alternative datasets. By referencing these external evaluations, we place our performance in a broader context, demonstrate how the observed trends align or diverge from established findings, and provide evidence that the advantages of Adaptive DeSeTra are not merely an artifact of a single dataset or experimental configuration but reflect more generalizable behavior across related security assessment settings. Building on this motivation, Table 6 summarizes representative state-of-the-art approaches from two complementary perspectives: a) the threat categories each method explicitly targets, and b) the corresponding performance settings reported in the original studies.

As shown in Table 6a, most existing systems are designed to cover benign activity together with a single adversarial family, most commonly prompt injection, while jailbreak detection is less consistently supported. Only a small subset of methods, such as JailGuard, explicitly targets all three categories, which limits the comparability of many reported results in terms of threat-model breadth. Importantly, Adaptive DeSeTra not only provides unified coverage across benign prompts, jailbreak attempts, and prompt-injection attacks within a single classifier, but also substantially outperforms Jail-

(a) Comparative analysis with state-of-the-art models across detection categories.

Model	Benign	Jailbreak	Prompt Inj.
Logistic Regression + BERT embeddings [13]	✓	✗	✓
JailbreakTracker (RoBERTa) [14]	✓	✓	✗
JailbreakTracker (BERT) [14]	✓	✓	✗
Random Forest + OpenAI embeddings [22]	✓	✗	✓
Random Forest (Snowflake) [16]	✓	✓	✗
RNN + BERT embeddings [23]	✓	✗	✓
Palisade [24]	✓	✗	✓
JailGuard (GPT-3.5-Turbo) [20]	✓	✓	✓
XLM-RoBERTa (zero-shot) [25]	✓	✗	✓
XLM-RoBERTa (finetuned) [25]	✓	✗	✓
LSTM [26]	✓	✓	✗
Simple Neural Network [26]	✓	✓	✗
Adaptive DeSeTra	✓	✓	✓

(b) Comparative analysis with state-of-the-art models on malicious prompt detection.

Model	Acc.	F1	Prec.	Rec.	AUC
Logistic Regression + BERT embeddings [13]	96.55	96.55	**100**	93.33	–
JailbreakTracker (RoBERTa) [14]	97.25	97.22	98.18	96.28	–
JailbreakTracker (BERT) [14]	96.74	96.76	96.03	97.50	–
Random Forest + OpenAI embeddings [22]	–	86.80	86.70	87.0	76.40
Random Forest (Snowflake) [16]	–	83.33	–	–	–
RNN + BERT embeddings [23]	94.74	–	–	–	–
Palisade [24]	88.78	86.75	78.26	97.30	–
JailGuard (GPT-3.5-Turbo) [20]	86.14	83.25	80.58	86.10	–
XLM-RoBERTa (zero-shot) [25]	55.17	62.32	55.13	71.67	–
XLM-RoBERTa (finetuned) [25]	99.14	99.16	**100**	98.33	–
LSTM [26]	91.69	69.0	75.0	63.0	–
Simple Neural Network [26]	92.52	73.0	75.0	72.0	–
Adaptive DeSeTra	**99.35**	**99.35**	99.35	**99.35**	**99.85**

Table 6: Comparison of detection coverage and test-set performance against state-of-the-art approaches.

Guard across the reported metrics. Specifically, JailGuard reports 86.14% Accuracy, 83.25% F1-score, 80.58% Precision, and 86.10% Recall, whereas Adaptive DeSeTra achieves 99.35% for Accuracy, F1-score, Precision, and Recall, together with an AUC of 99.85%, demonstrating a markedly stronger and more reliable detection capability within a comprehensive operational setting.

On the other hand, Table 6b complements the coverage analysis by summarizing the performance figures reported in prior studies for malicious prompt detection, using the metrics available in each work. Adaptive DeSeTra emerges as the strongest overall choice because it delivers the highest, most consistent performance across the full metric suite under a unified objective. Adaptive DeSeTra achieves the highest reported Accuracy, F1-score, Recall, and AUC, reaching 99.35% for Accuracy and F1 and 99.85% AUC, which indicates both near-perfect classification decisions and excellent separability between benign and malicious inputs. This combination is important because it shows not only a very high rate of correct decisions at the selected operating point,

but also excellent separability between classes across thresholds, as reflected by the AUC. The fact that Precision and Recall are both high indicates that Adaptive DeSeTra maintains a well-balanced error profile, minimizing both false alarms and missed detections. Overall, these results suggest that Adaptive DeSeTra provides highly reliable and stable discrimination capability, which is essential for security-oriented deployment where both attack coverage and consistent decision quality are required.

Although Multilingual BERT [13] and fine-tuned XLM-RoBERTa [25] report slightly higher Precision, their performance profiles are less balanced, and their evaluation scope is narrower. Specifically, Multilingual BERT attains perfect Precision at the cost of lower Accuracy, F1-score, and Recall. Similarly, fine-tuned XLM-RoBERTa, despite its high Precision, still reports lower Accuracy, F1-score, and Recall than Adaptive DeSeTra. A further limitation is that both approaches are validated on a single benchmark, the DeepSet Prompt Injection Dataset [21], using its predefined split of 546 training prompts and 116 test prompts. Reliance on one relatively small, fixed corpus limits conclusions about robustness and generalization. Results may reflect dataset-specific artifacts such as prompt phrasing, labeling conventions, or narrow attack patterns, and may not transfer to other injection styles, domains, or evolving adversarial strategies encountered in deployment. In addition, both works frame the task as binary classification, distinguishing benign prompts from prompt-injection attempts, rather than jointly separating multiple adversarial families. In practice, this Precision-heavy behavior typically reflects a stricter operating threshold that suppresses false positives but allows more attacks to pass undetected, which is undesirable in security screening, and it is not measured under the same multi-class, multi-threat setting.

In contrast, Adaptive DeSeTra optimizes a single model for simultaneous discrimination among benign prompts, jailbreak attempts, and prompt-injection attacks, and achieves the strongest results across Accuracy, F1-score, Recall, and AUC. Moreover, Adaptive DeSeTra is the only approach that reports strong results across all core discrimination metrics and provides an AUC value, supporting the conclusion that it generalizes robustly and is the most suitable deployment model when both broad threat coverage and balanced error behavior are required. In addition, Adaptive DeSeTra is trained at a substantially larger scale, using 164,749 prompts for training, 20,517 for validation, and 20,520 for testing, with non-overlapping splits such that each partition contains unique prompts and no instances are

Dataset	Model	Benign	Jailbreak	Prompt Inj.	Total Prompts
JailbreakBench/JBB-Behaviors	Vicuna, Mistral, Llama-2, Llama-3	100	100	0	200
neuralchemy/Prompt-injection-dataset	–	2,538	389	2,061	4,988

Table 7: Datasets used for cross-dataset generalization evaluation in prompt classification

shared across training, validation, and testing. This gives Adaptive DeSeTra an advantage because it reduces the risk of memorization or leakage-driven inflation of test performance and provides a more reliable estimate of how the model will generalize to unseen, real-world prompt distributions.

4.1.7. Cross-dataset generalization results

To determine whether the prompt-classification results remain stable on independent data, we next extend the evaluation to Adaptive DeSeTra and all baseline models on external corpora that were not involved in training, validation, or testing. To this end, Table 7 summarizes the two datasets employed for cross-dataset generalization evaluation. The first, `JailbreakBench/JBB-Behaviors` [30] dataset, contains 200 prompts generated across multiple LLM families, including Vicuna, Mistral, Llama-2, Llama-3, and provides a balanced setting with 100 `Benign` and 100 `Jailbreak` samples, but no `Prompt Injection` instances. The second, `neuralchemy/Prompt-injection-dataset`, provides a larger and more heterogeneous benchmark of 4,988 prompts. Because the original corpus includes attack categories beyond the scope of this study, we retained only the instances relevant to our three-class setting, namely 2,538 `Benign`, 389 `Jailbreak`, and 2,061 `Prompt Injection` prompts. In particular, the `Prompt Injection` subset was formed by aggregating prompts labeled as `direct_injection`, `indirect_injection`, `instruction_override`, and `prompt_injection`, yielding 2,061 prompts in total.

Table 8 shows the evaluation metrics obtained by Adaptive DeSeTra and all baseline models across the two independent datasets. As shown in Table 8a, Adaptive DeSeTra achieves the strongest overall performance, reaching 99.50% for Accuracy, F1-score, Precision and Recall, together with 99.91% AUC. The closest baselines are DeBERTa-v2 and XLNet, both at 98.50% on the primary metrics, indicating that Adaptive DeSeTra preserves a clear performance advantage even on this compact external benchmark. Similarly, Table 8b shows that Adaptive DeSeTra again attains the highest results under a larger and more heterogeneous evaluation setting, achiev-

(a) Prompt classification performance on the JailbreakBench/JBB-Behaviors dataset.

Model	Acc. (%)	F1 (%)	Prec. (%)	Rec. (%)	AUC (%)
Albert	98.00	98.00	98.00	98.00	98.23
Bert	95.00	95.00	95.00	95.00	98.82
Deberta-v2	98.50	98.50	98.50	98.50	99.34
Distilbert	97.00	97.00	97.00	97.00	99.91
Electra	96.50	96.50	96.50	96.50	99.17
GPT-2	97.00	97.00	97.00	97.00	99.52
Longformer	98.00	98.00	98.00	98.00	99.76
RoBERTa	95.50	95.50	95.50	95.50	99.10
XLCNN	98.00	98.00	98.00	98.00	99.23
XLNet	98.50	98.50	98.50	98.50	99.79
Adaptive DeSeTra	**99.50**	**99.50**	**99.50**	**99.50**	**99.91**

(b) Prompt classification performance on the neuralchemy/Prompt-injection-dataset dataset.

Model	Acc. (%)	F1 (%)	Prec. (%)	Rec. (%)	AUC (%)
Albert	96.93	96.54	96.81	96.28	97.47
Bert	96.11	95.62	95.91	95.33	98.11
Deberta-v2	98.32	98.13	98.26	98.00	99.25
Distilbert	98.26	98.05	98.22	97.88	99.01
Electra	97.81	97.51	97.68	97.35	98.64
GPT-2	96.55	96.08	96.39	95.79	97.76
Longformer	98.12	97.87	98.04	97.71	99.12
RoBERTa	97.31	96.97	97.17	96.77	98.02
XLCNN	98.44	98.22	98.35	98.08	99.39
XLNet	97.83	97.53	97.69	97.36	98.92
Adaptive DeSeTra	**99.39**	**99.31**	**99.35**	**99.28**	**99.86**

Table 8: Cross-dataset generalization results for prompt classification

ing 99.39% Accuracy, 99.31% F1-score, 99.35% Precision, 99.28% Recall and 99.86% AUC. In this case, XLCNN is the strongest baseline with 98.44% Accuracy and 98.22% F1-score, but it still remains below Adaptive DeSeTra across all reported metrics. Overall, while the relative positions of the baseline models shift across the two external datasets, Adaptive DeSeTra remains consistent, reinforcing its robustness under cross-dataset generalization.

4.2. Impact Assessment

Building on the prompt-classification results above, where Adaptive DeSeTra demonstrates reliable and well-balanced discrimination among Benign, Jailbreak and Prompt Injection inputs under both internal baselines and external comparisons, the next step is to move from intent detection to impact assessment. In operational terms, flagging a malicious prompt is necessary but not sufficient, since the security consequence ultimately depends on whether the targeted LLM complies with the adversarial instruction or refuses it. Accordingly, the evaluation is extended with an impact-assessment component that measures attack effectiveness directly from the LLM output, by training Adaptive DeSeTra to classify responses into `Compliant` or `Refusal`, thereby enabling a quantitative, response-level success or failure determination for Jailbreak and Prompt Injection attempts.

4.2.1. Proposed Dataset for Impact Assessment

Adaptive DeSeTra extends beyond malicious prompt detection by assessing whether an input actually influences the target LLM, based on the response it produces. This capability forms a second security layer that evaluates the outcome of a request at the response level. Importantly, the impact-

Dataset	Model	Compliant	Refusal	Total Responses
tatsu-lab/alpaca	text-davinci-003	52,002	0	52,002
nothingiisreal/Claude-3-Opus-Instruct-15K	Claude-3-opus	13,669	0	13,669
Dahoas/instruct-synthetic-prompt-responses	–	33,201	0	33,201
LLM-Detect-AI-Generated-Text-Dataset	–	11,637	0	11,637
andrewsiah/personalization_prompt_response_llama3_sfairx	GPT-4-turbo, GPT-4o, Claude-3-opus, Llama-3-70b-instruct	10,431	0	10,431
sequelbox/Celestia3-DeepSeek-R1-0528	DeepSeek-R1-0528	90,990	0	90,990
vaidehi99/LC_concepts_refusal	–	0	1,053	1,053
Human-CentricAI/chatbot-arena-llm-refusal	Chatbot arena	2,284	594	2,878
avrecum/refusals_dataset_ultra2	Claude, Grok	0	518,735	518,735
mrfakename/refusal-xl	TinyLlama-1.1B	0	16,006	16,006
Proposed	All models above	208,126	470,650	678,776

Table 9: Proposed dataset for training the Adaptive DeSeTra to distinguish between compliant and refusal LLM responses

assessment task is intentionally independent of the original prompt label; its purpose is to determine whether the LLM ultimately complied with or refused the issued request, rather than to reclassify the prompt as adversarial or benign. Accordingly, the labeling criterion is based on the final behavioral outcome of the LLM output. A response is labeled as `Compliant` if it substantively fulfills the user instruction or provides the requested actionable content, even when accompanied by warnings, hedging, or brief safety disclaimers. In contrast, a response is labeled as `Refusal` if it declines, blocks, withholds, or redirects the request in a way that prevents actual fulfillment. During dataset construction, responses from the selected public corpora were harmonized into this binary taxonomy. For ambiguous cases, the decisive factor was whether the response ultimately delivered the requested content, rather than the superficial tone of the opening sentence.

Table 9 summarizes the dataset used to train Adaptive DeSeTra for impact assessment. It contains 678,776 responses split into two classes: 208,126 `Compliant` and 470,650 `Refusal`. The compliant class was built from multiple instruction–response datasets and includes outputs from models such as text-davinci-003, Claude-3-opus, GPT-4-turbo, GPT-4o, Llama-3-70b-instruct, DeepSeek-R1-0528 and Chatbot Arena. The refusal class was built from datasets containing explicit denials generated by systems such as Claude, Grok, TinyLlama-1.1B and Chatbot Arena. After removing duplicates and non-English entries, the final corpus provides broad coverage of both instruction-following and refusal behavior.

Class	Train	Validation	Testing
`Compliant`	166,501	20,812	20,813
`Refusal`	429,025	20,812	20,813
`Total Responses`	595,526	41,624	41,626

Table 10: Dataset splitting into training, validation, and testing subsets for impact assessment

4.2.2. Preprocessing of the Impact Assessment Dataset

Before partitioning the response-level corpus, the collected LLM outputs underwent a preprocessing pipeline to reduce redundancy and prevent data leakage across training, validation, and testing. All responses gathered from the ten source repositories were normalized to a common textual format, de-duplicated within and across sources, and filtered to retain only English entries. This step was especially important for the Refusal class, where repeated safety disclaimers, policy notices, and near-identical denial templates are common. In total, 65,738 Refusal samples were removed because they were either non-English or duplicates. Such filtering was necessary to avoid overfitting to a limited set of highly repetitive refusal formulations.

Beyond exact duplicate removal, an additional similarity analysis was performed to provide a quantitative estimate of intra-class redundancy and potential overlap within the dataset. Each response was represented using TF–IDF vectors with unigram and bigram features, and cosine similarity was computed separately for the `Compliant` and `Refusal` classes. The mean cosine-similarity was 0.14 for `Compliant` and 0.17 for `Refusal`. These relatively low values indicate limited lexical redundancy within both classes after preprocessing. Although Refusal responses exhibit slightly higher similarity, likely due to recurring safety-oriented phrasing, the overall overlap remains low, indicating that the dataset is not dominated by repetitive response templates. After this preprocessing stage, the resulting corpus was partitioned into three mutually exclusive subsets, namely training, validation, and testing, as reported in Table 10. The split was performed at the response level, ensuring exclusive assignment of each instance to a single subset and preventing duplicated or overlapping samples, thereby supporting an objective and reproducible evaluation protocol.

4.2.3. Training process

Following preprocessing, the resulting dataset was partitioned as shown in Table 10. The training subset contains 166,501 compliant responses and

429,025 refusal responses, for a total of 595,526 samples. The validation subset includes 20,812 instances per class, yielding 41,624 responses in total. Likewise, the test subset contains 20,813 instances per class, corresponding to 41,626 responses overall. Although the raw training pool is class-imbalanced, balanced optimization was deliberately enforced through stochastic sampling in order to prevent bias toward the majority `Refusal` class and to ensure that the learned decision boundary reflects objective discriminative ability rather than class prevalence. Specifically, across 10 epochs, each epoch randomly selected 40,000 unique instances per class from the training set, ensuring that both labels contributed equally to parameter updates, improving stability and preventing the `Refusal` class from dominating the gradient signal. For the same reason, the validation and test sets were also evaluated on balanced splits, so that the reported results reflect unbiased classification performance across `Compliant` and `Refusal` responses rather than being inflated by deployment-specific class priors. Finally, the training, validation and test splits were constructed to be mutually exclusive, ensuring no overlap and preventing data leakage across phases.

Similar to the prompt-classification experiments, we identified the most suitable configuration for impact assessment by training multiple Transformer architectures under the same procedure and comparing their learning dynamics using cumulative average training loss. As depicted in Figure 4, all models exhibit a decreasing loss trend, with the largest improvements occurring early in training and smaller gains as optimization progresses. However, the convergence speed and final loss levels differ clearly across architectures. Adaptive DeSeTra reaches the lowest cumulative training loss and remains the most stable model throughout training (0.0251). All baselines converge to higher loss regions, with RoBERTa (0.0414), XLCNN (0.0424), DeBERTa-v2 (0.0541), and DistilBERT (0.0613) forming the strongest baseline tier, followed by GPT-2 (0.0731), Longformer (0.1208), ALBERT (0.1211), ELECTRA (0.1450), XLNet (0.1630), and BERT (0.2093). Overall, these results show that Adap-

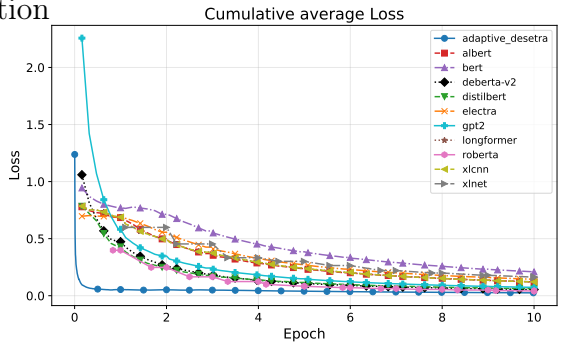
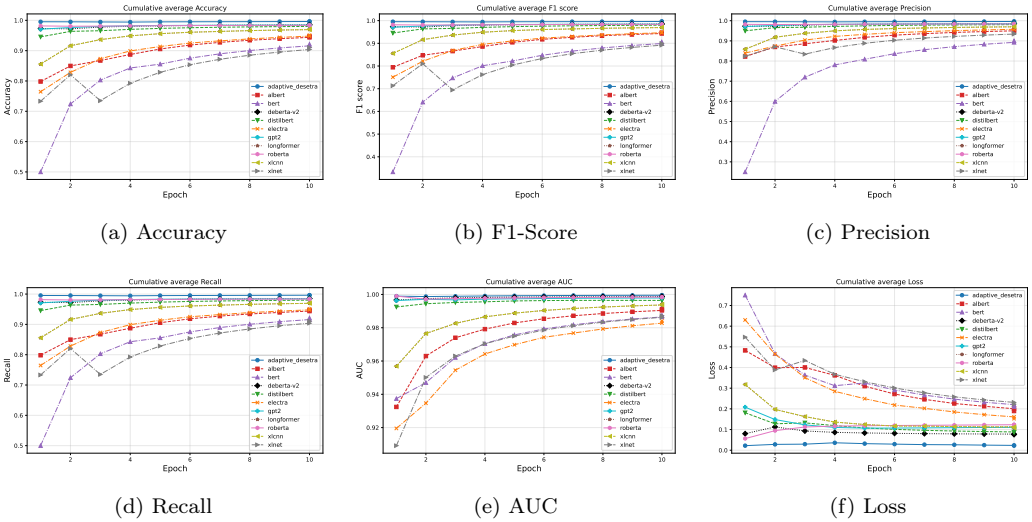


Figure 4: Cumulative average training loss across Transformer models for impact assessment



(a) Accuracy (b) F1-Score (c) Precision

(d) Recall (e) AUC (f) Loss

Figure 5: Validation curves reported as cumulative averages across epochs for different Transformer models on impact assessment.

tive DeSeTra learns the compliant-versus-refusal decision boundary more efficiently than all evaluated baselines, supporting its selection as the best-performing configuration for response-level impact assessment.

4.2.4. Validation results

Building on the training dynamics discussed above, the next step is to quantify how well each model generalizes to unseen Compliant and Refusal responses. Accordingly, after each training epoch, each Transformer was evaluated on the validation set using the same metric suite used for prompt classification. Figure 5 reports these validation measurements as cumulative averages across epochs, which emphasizes stable generalization over isolated single-epoch peaks and therefore provides a reliable basis for model selection.

Across all validation metrics, Adaptive DeSeTra demonstrates the strongest and most consistent overall performance, attaining the highest values in Accuracy, F1-score, Precision and Recall (99.82%), as well as the highest AUC (99.99%) and the lowest Loss (0.0112). Among the baselines, DeBERTa-v2 emerges as the closest competitor, achieving 98.95% Accuracy, F1-score, and Precision, together with a 99.89% AUC and a Loss of 0.0537. A second high-performing group includes GPT-2 (98.77% Accuracy, 99.61% AUC, 0.0711 Loss), XLCNN (98.75% Accuracy, 99.77% AUC, 0.0638 Loss), DistilBERT (98.67% Accuracy, 99.70% AUC, 0.0684 Loss), and Longformer (98.64% Ac-

(a) Adaptive DeSeTra test-set confusion statistics for impact assessment.

Class	TP	FP	TN	FN	FPR	FNR
Compliant	20,759	31	20,782	54	0.00149	0.00259
Refusal	20,782	54	20,759	31	0.00259	0.00149
Overall	41,541	85	41,541	85	0.00204	0.00204

(b) Adaptive DeSeTra test-set performance for impact assessment.

Class	Acc.	F1	Prec.	Rec.
Compliant	99.80	99.80	99.85	99.74
Refusal	99.80	99.80	99.74	99.85
Overall	99.80	99.80	99.80	99.80

Table 11: Adaptive DeSeTra results on the impact assessment test set, summarizing confusion matrix and per-class performance metrics.

curacy, 99.84% AUC, 0.0571 Loss), all of which show strong validation behavior yet remain consistently below Adaptive DeSeTra. Slightly behind this group are RoBERTa (98.54% Accuracy, 99.88% AUC, 0.1256 Loss), ELECTRA (98.47% Accuracy, 99.69% AUC, 0.0633 Loss), and ALBERT (98.38% Accuracy, 99.79% AUC, 0.0802 Loss), which form the next performance tier. By contrast, XLNet (97.97% Accuracy, 99.77% AUC, 0.1249 Loss) and BERT (97.76% Accuracy, 99.75% AUC, 0.1131 Loss) yield the weakest validation results among the evaluated baselines. Taken together, these findings confirm that Adaptive DeSeTra generalizes more effectively and more reliably than all competing models.

4.2.5. Testing results

Having identified the most reliable configuration based on validation stability and cumulative performance trends, the final step is to confirm that these gains persist under a fully isolated evaluation. Adaptive DeSeTra is thus assessed once on the test split, which remained strictly unseen during both training and validation, providing an unbiased estimate of out-of-sample generalization for response-level impact assessment. Table 11a reports the class-wise confusion statistics of Adaptive DeSeTra on the isolated impact-assessment test set, with the derived false positive and false negative rates. The counts confirm that misclassifications are rare relative to the number of evaluated responses. For `Compliant`, the model produces 20,759 true positives and only 31 false positives, with 54 false negatives, yielding low error rates of FPR = 0.00149 and FNR = 0.00259. For `Refusal`, the pattern is similarly strong, with 20,782 true positives, 54 false positives, and 31 false negatives, corresponding to FPR = 0.00259 and FNR = 0.00149. The "Overall" row aggregates these outcomes across both classes and indicates a balanced residual error profile, with 85 false positives and 85 false nega-

tives in total and matched average rates of FPR = FNR = 0.00204, which is desirable for impact assessment because it avoids a directional tendency to over-label outputs as Compliant or as Refusal.

Complementing the error-rate view, Table 11b shows that Adaptive DeSeTra achieves uniformly high performance for both `Compliant` and `Refusal` responses, reaching 99.80% Accuracy and F1-score across classes and overall, which indicates highly reliable and well-balanced decisions. The class-wise Precision and Recall values are consistent with the confusion matrix, with `Compliant` exhibiting slightly higher Precision than Recall and `Refusal` showing the inverse pattern, as expected from their residual false positive/false negative trade-off. A qualitative inspection of the few remaining misclassifications shows that they are concentrated in borderline cases where refusal and compliance signals co-exist in the same output, such as responses that begin with safety-oriented or refusal-style language but later provide partial actionable assistance, or responses that avoid explicit denial markers while still withholding the requested content through indirect redirection. These findings confirm that the most difficult cases are mixed or weakly signaled responses in which the surface phrasing does not fully align with the final behavioral outcome, reinforcing the need for explicit response-level impact assessment.

4.2.6. Comparative Analysis

To compare the effectiveness and efficiency of Adaptive DeSeTra for response-level impact assessment, we evaluated the same set of Transformer baselines under an identical testing protocol. Specifically, we report (*i*) discriminative performance on the test set using Accuracy, F1-score, Precision, Recall and AUC, and (*ii*) the corresponding average inference-time resource footprint observed during the test phase, allowing performance gains to be interpreted alongside practical deployment cost. To the best of our knowledge, prior research that analyzes compliance versus refusal typically treats it as a supporting signal within broader safety or moderation pipelines, rather than training and reporting a dedicated classifier whose sole objective is to label LLM outputs for impact assessment. As a consequence, directly comparable benchmark results and standardized metrics for this response-level task are generally not available in the literature. Therefore, we position our evaluation as an internal comparative study across widely used Transformer models, all fine-tuned under the same training procedure, to ensure that the reported differences in outcomes are attributable to the model architectures rather

(a) Test-set performance comparison across Transformer models on the proposed dataset.

Model	Acc.	F1	Prec.	Rec.	AUC
Albert	98.38	98.38	98.38	98.38	99.79
Bert	97.63	97.63	97.65	97.63	99.73
Deberta-v2	98.95	98.95	98.95	98.95	99.87
Distilbert	98.62	98.62	98.63	98.62	99.72
Electra	98.37	98.37	98.37	98.37	99.70
GPT-2	98.66	98.66	98.66	98.66	99.83
Longformer	98.61	98.61	98.61	98.61	99.84
RoBERTa	98.54	98.54	98.54	98.54	99.87
XLCNN	98.75	98.75	98.75	98.75	99.77
XLNet	97.83	97.83	97.83	97.83	99.75
Adaptive DeSeTra	**99.80**	**99.80**	**99.80**	**99.80**	**99.98**

(b) Average system resource usage during model inference for impact assessment.

Model	Params (Mil.)	CPU (%)	RAM (MB)	VRAM (MB)	Pred. Time (s)
Albert	13	0.244	44	578	0.0062
Bert	109	0.348	0	980	0.0102
Deberta-v2	313	0.236	2	1752	0.0061
Distilbert	67	0.229	0	800	0.0060
Electra	14	0.326	0	582	0.0118
GPT-2	124	0.262	28	1054	0.0108
Longformer	359	0.324	450	1998	0.0405
RoBERTa	355	0.261	404	1914	0.0223
XLCNN	331	0.259	127	1825	0.0125
XLNet	342	0.315	420	1885	0.0291
Adaptive DeSeTra	323	0.255	380	1794	0.0066

Table 12: Impact assessment comparison across Transformer models, reporting test-set performance and average inference-time resource usage.

than to experimental inconsistencies.

The test-set performance of the Transformer baselines and Adaptive DeSeTra for response-level impact assessment is summarized in Table 12a. Across the full set of evaluation metrics, Adaptive DeSeTra achieves the strongest overall performance, attaining 99.80% for Accuracy, F1-score, Precision and Recall, together with the highest AUC of 99.98%. Among the baselines, DeBERTa-v2 emerges as the closest competitor, reaching 98.95% across Accuracy, F1-score, Precision and Recall, with an AUC of 99.87%. It is followed by XLCNN (98.75%, 99.77% AUC), GPT-2 (98.66%, 99.83% AUC), DistilBERT (98.62%, 99.72% AUC) and Longformer (98.61%, 99.84% AUC), all of which deliver strong test-set results but remain below Adaptive DeSeTra. RoBERTa (98.54%, 99.87% AUC), Albert (98.38%, 99.79% AUC) and Electra (98.37%, 99.70% AUC) form the next performance tier, whereas XLNet (97.83%, 99.75% AUC) and Bert (97.63%, 99.73% AUC) yield the weakest results among the evaluated baselines.

Table 12b reports the average inference-time resource footprint during testing, characterizing the deployment cost of each model as a response-level judge. Measurements were obtained on an NVIDIA A100-SXM4 GPU with batch size 32, and prediction times include tokenization overhead. Overall, CPU utilization remains very low for all models (0.229%–0.348%), while memory usage scales with model size. Albert and Electra have the smallest parameter counts and lowest VRAM usage, whereas larger models such as

Model	Acc. 95% Wilson CI	Prec. 95% Wilson CI	Rec. 95% Wilson CI
Albert	[98.25, 98.50]	[98.25, 98.50]	[98.25, 98.50]
Bert	[97.48, 97.77]	[97.48, 97.77]	[97.48, 97.77]
Deberta-v2	[98.85, 99.04]	[98.85, 99.04]	[98.85, 99.04]
Distilbert	[98.50, 98.73]	[98.50, 98.73]	[98.50, 98.73]
Electra	[98.24, 98.49]	[98.24, 98.49]	[98.24, 98.49]
GPT-2	[98.54, 98.77]	[98.54, 98.77]	[98.54, 98.77]
Longformer	[98.49, 98.72]	[98.49, 98.72]	[98.49, 98.72]
RoBERTa	[98.42, 98.65]	[98.42, 98.65]	[98.42, 98.65]
XLCNN	[98.64, 98.85]	[98.64, 98.85]	[98.64, 98.85]
XLNet	[97.69, 97.97]	[97.69, 97.97]	[97.69, 97.97]
Adaptive DeSeTra	**[99.75, 99.84]**	**[99.75, 99.84]**	**[99.75, 99.84]**

Table 13: 95% Wilson score confidence intervals for overall Accuracy, Precision and Recall on impact assessment.

Dataset	Model	Compliant	Refusal	Total Responses
`sorry-bench/sorry-bench-human-judgment-202503`	GPT-4o, GPT-3.5-turbo, Llama-3-8b, Llama-3-70b, Gemma-7b, Vicuna-7b, Mistral-7b, OpenChat-3.5	2,139	4,901	7,040
Synthetic data	GPT-4o-mini, DeepSeek-V3.2, Llama-3.1-8b	1,500	1,500	3,000

Table 14: Datasets used for cross-dataset generalization evaluation in impact assessment

Longformer, RoBERTa, XLNet, XLCNN, Deberta-v2 and Adaptive DeSeTra require substantially more accelerator memory. In terms of latency, Distilbert and Deberta-v2 are the fastest, with Albert and Adaptive DeSeTra close behind, while Longformer, XLNet and RoBERTa are the slowest.

To further quantify the reliability of these comparisons, we additionally report 95% Wilson score confidence intervals for each model's impact-assessment Accuracy, Precision and Recall, thereby capturing the uncertainty associated with finite test-set size. As shown in Table 13, Adaptive DeSeTra remains clearly separated from the strongest baseline, with non-overlapping confidence intervals. Overall, this separation indicates that the advantage of Adaptive DeSeTra in response-level impact assessment is statistically robust, supporting its selection as the most effective configuration for distinguishing `Compliant` from `Refusal` responses.

4.2.7. Cross-dataset Generalization Results

To determine whether the impact-assessment results remain stable on independent data, we next extend the evaluation to Adaptive DeSeTra and all

(a) Response-level classification performance on the sorry-bench/sorry-bench-human-judgment-202503 dataset.

Model	Acc. (%)	F1 (%)	Prec. (%)	Rec. (%)	AUC (%)
Albert	98.45	98.18	98.00	98.36	99.67
Bert	97.14	96.65	96.34	96.97	98.24
Deberta-v2	98.78	98.56	98.42	98.70	99.89
Distilbert	98.72	98.49	98.34	98.65	99.43
Electra	98.34	98.04	97.85	98.24	99.63
GPT-2	97.57	97.15	96.88	97.43	98.87
Longformer	98.69	98.46	98.31	98.61	99.88
RoBERTa	98.66	98.43	98.28	98.58	99.74
XLCNN	98.81	98.59	98.46	98.73	99.79
XLNet	97.95	97.59	97.37	97.83	99.78
Adaptive DeSeTra	**99.77**	**99.73**	**99.71**	**99.76**	**99.90**

(b) Response-level classification performance on the dataset created by GPT-4o-mini, DeepSeek-V3.2, Llama3.1-8b.

Model	Acc. (%)	F1 (%)	Prec. (%)	Rec. (%)	AUC (%)
Albert	98.13	98.13	98.13	98.13	99.74
Bert	97.53	97.53	97.53	97.53	99.33
Deberta-v2	99.11	99.11	99.11	99.11	99.89
Distilbert	98.46	98.46	98.46	98.46	99.75
Electra	98.42	98.42	98.42	98.42	99.66
GPT-2	98.87	98.87	98.87	98.87	99.81
Longformer	98.22	98.22	98.22	98.22	99.76
RoBERTa	98.19	98.19	98.19	98.19	99.71
XLCNN	99.02	99.02	99.02	99.02	99.86
XLNet	97.89	97.89	97.89	97.89	99.45
Adaptive DeSeTra	**99.86**	**99.86**	**99.86**	**99.86**	**99.94**

Table 15: Cross-dataset generalization results for impact assessment

baseline models on external corpora that were not involved in training, validation, or testing. To this end, Table 14 summarizes the two datasets employed for cross-dataset generalization evaluation. The first, `sorry-bench/sorry-bench-human-judgment-202503` [31], contains 7,040 responses collected from multiple LLM families, and provides a realistic evaluation setting with 2,139 `Compliant` and 4,901 `Refusal` responses. The second dataset is synthetic and was constructed because, to the best of our knowledge, no other publicly available dataset exists for response-level cross-dataset evaluation under the same `Compliant`/`Refusal` formulation. Specifically, GPT-4o-mini, DeepSeek-V3.2 and Llama3.1-8b were each used to generate 500 `Compliant` and 500 `Refusal` responses, yielding a balanced corpus of 1,500 samples per class and 3,000 responses in total.

Table 15 reports the evaluation metrics obtained by Adaptive DeSeTra and all baseline models across the two independent datasets. As shown in Table 15a, Adaptive DeSeTra achieves the strongest overall performance, reaching 99.77% Accuracy, 99.73% F1-score, 99.71% Precision, 99.76% Recall, and 99.90% AUC. The closest baseline is XLCNN, which attains 98.81% Accuracy and 98.59% F1-score, indicating that Adaptive DeSeTra preserves a clear performance advantage on this external benchmark. Likewise, Table 15b shows that Adaptive DeSeTra also attains the highest results on the synthetic dataset, achieving 99.86% across Accuracy, F1-score, Precision and Recall, together with 99.94% AUC. In this case, DeBERTa-v2 is the strongest baseline with 99.11% on the primary metrics, followed closely by XLCNN at 99.02%, yet both remain below Adaptive DeSeTra.

5. Limitations and Future Work

Despite its advantages, Adaptive DeSeTra has deployment-relevant limitations: inference averages 371–380 MB RAM and about 1,794 MB VRAM across both tasks, which can reduce service density and require sufficiently capable GPUs. It is also English-only, limiting language coverage. Future work will focus on improving Adaptive DeSeTra's deployability through architectural and systems-level optimizations that reduce RAM/VRAM without sacrificing latency or accuracy. We also plan to extend the model to multilingual text, evaluate it under more realistic operational conditions, including dynamic real-world deployment scenarios and study deployment scalability in distributed LLM infrastructures. In addition, we plan to incorporate an adversarial prompt-generation module for dataset expansion and continual hardening against emerging jailbreak and prompt-injection strategies.

6. Conclusion

In conclusion, Adaptive DeSeTra provides a unified and effective component for LLM security by combining prompt-level intent detection with response-level impact assessment in a single model family. Through its adaptive deformable-span self-attention mechanism and encoder-level convolutional augmentation, it achieves strong discrimination across both malicious prompt types and response outcomes, while remaining robust on independent cross-dataset evaluations. Experimental results show that Adaptive DeSeTra consistently outperforms a broad set of Transformer baselines on both prompt classification and impact assessment, confirming its effectiveness as a security-focused model for analyzing adversarial prompts and LLM responses. Overall, the proposed approach advances beyond binary prompt filtering by coupling fine-grained threat attribution with direct measurement of attack success, thereby offering a more complete basis for protecting and evaluating LLM-based systems.

Acknowledgments

This research has received funding from the European Union's Horizon Europe program under Grant Agreement No. 101131292 (AIAS), from the Digital Europe Programme under Grant Agreement No. 101145872 (NITRO) and No. 101128024 (CyberUnity), and from the European Cybersecurity Competence Centre under Grant Agreement No. 101168407 (CPAID).

References

[1] H. Naveed, A. U. Khan, S. Qiu, M. Saqib, S. Anwar, M. Usman, N. Akhtar, N. Barnes, A. Mian, A comprehensive overview of large language models, ACM Transactions on Intelligent Systems and Technology 16 (2025) 1–72.

[2] Z. Zheng, K. Ning, Q. Zhong, J. Chen, W. Chen, L. Guo, W. Wang, Y. Wang, Towards an understanding of large language models in software engineering tasks, Empirical Software Engineering 30 (2025) 50.

[3] D. B. Acharya, K. Kuppan, B. Divya, Agentic ai: Autonomous intelligence for complex goals–a comprehensive survey, IEEe Access (2025).

[4] M. Abou Ali, F. Dornaika, J. Charafeddine, Agentic ai: a comprehensive survey of architectures, applications, and future directions, Artificial Intelligence Review 59 (2025) 11.

[5] F. Adjewa, M. Esseghir, L. Merghem-Boulahia, C. Kacfah, Llm-based continuous intrusion detection framework for next-gen networks, in: 2025 International Wireless Communications and Mobile Computing (IWCMC), IEEE, 2025, pp. 1198–1203.

[6] R. Sharma, A. Gupta, Source code guardrail: Ai driven solution to distinguish critical vs. generic code for enterprise llm security, in: International Conference on Provable Security, Springer, 2025, pp. 449–458.

[7] F. He, T. Zhu, D. Ye, B. Liu, W. Zhou, P. Yu, The emerged security and privacy of llm agent: A survey with case studies, ACM Computing Surveys (2024).

[8] Y. Xue, J. Wang, Z. Yin, Y. Ma, H. Qin, R. Tao, X. Liu, Dual intention escape: Penetrating and toxic jailbreak attack against large language models, in: Proceedings of the ACM on Web Conference 2025, 2025, pp. 863–871.

[9] M. Bombieri, S. P. Ponzetto, M. Rospocher, The dangerous effects of a frustratingly easy llms jailbreak attack, IEEE Access 13 (2025). doi:10.1109/ACCESS.2025.3589112.

[10] Z. Shao, H. Liu, J. Mu, N. Z. Gong, Enhancing prompt injection attacks to llms via poisoning alignment, in: Proceedings of the 2025 ACM Workshop on Artificial Intelligence and Security (AISec '25), ACM, 2025, pp. 13–27. doi:10.1145/3733799.3762963.

[11] S. Dora, D. Lunkad, N. Aslam, S. Venkatesan, S. K. Shukla, The hidden risks of llm-generated web application code: A security-centric evaluation of code generation capabilities in large language models, in: International Conference on Information Systems Security, Springer, 2025, pp. 27–37.

[12] T. Wu, H. Zhang, Chain-of-detection enables robust and efficient jailbreak defense, Neural Networks (2025) 108217.

[13] M. A. Rahman, H. Shahriar, F. Wu, A. Cuzzocrea, Applying pre-trained multilingual bert in embeddings for improved malicious prompt injection attacks detection, in: 2024 2nd International Conference on Artificial Intelligence, Blockchain, and Internet of Things (AIBThings), IEEE, 2024, pp. 1–7.

[14] M. F. A. Sayeedi, M. B. Hossain, M. K. Hassan, S. Afrin, M. M. Sabit, M. S. Hossain, Jailbreaktracer: Explainable detection of jailbreaking prompts in llms using synthetic data generation, IEEE Access (2025).

[15] M. A. Ferrag, N. Tihanyi, D. Hamouda, L. Maglaras, A. Lakas, M. Debbah, From prompt injections to protocol exploits: Threats in llm-powered ai agents workflows, ICT Express (Elsevier) (2025). doi:10.1016/j.icte.2025.12.001.

[16] E. Galinkin, M. Sablotny, Improved large language model jailbreak detection via pretrained embeddings, arXiv preprint arXiv:2412.01547 (2024).

[17] X. Shen, Z. Chen, M. Backes, Y. Shen, Y. Zhang, " do anything now": Characterizing and evaluating in-the-wild jailbreak prompts on large language models, in: Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security, 2024, pp. 1671–1685.

[18] L. Derczynski, E. Galinkin, S. Majumdar, Garak: A framework for large language model red teaming, Garak Ai–Available from: https://garak.ai (2024).

[19] Huggingface, jackhhao, jackhhao/jailbreak-classification, https://huggingface.co/datasets/jackhhao/jailbreak-classification, (Accessed: January 2026).

[20] X. Zhang, C. Zhang, T. Li, Y. Huang, X. Jia, M. Hu, J. Zhang, Y. Liu, S. Ma, C. Shen, Jailguard: A universal detection framework for prompt-based attacks on llm systems, ACM Transactions on Software Engineering and Methodology (2025).

[21] Huggingface, deepseet, deepset/prompt-injections, https://huggingface.co/datasets/deepset/prompt-injections, (Accessed: January 2026).

[22] M. A. Ayub, S. Majumdar, Embedding-based classifiers can detect prompt injection attacks, arXiv preprint arXiv:2410.22284 (2024).

[23] B. Jain, P. Pawar, D. Gada, T. Patwa, P. Kanani, D. Patil, L. Kurup, Comprehensive analysis of machine learning and deep learning models on prompt injection classification using natural language processing techniques, International Research Journal of Multidisciplinary Technovation 7 (2025) 24–37.

[24] S. Kokkula, G. Divya, et al., Palisade–prompt injection detection framework, arXiv preprint arXiv:2410.21146 (2024).

[25] M. A. Rahman, H. Shahriar, G. Francia, F. Wu, A. Cuzzocrea, M. Rahman, M. J. H. Faruk, S. I. Ahamed, Fine-tuned large language models (llms): Improved prompt injection attacks detection, in: 2025 IEEE 49th Annual Computers, Software, and Applications Conference (COMPSAC), IEEE, 2025, pp. 1033–1039.

[26] H. Deshpande, D. Chaudhari, T. Sarode, A. Kamath, Exploring llm and neural networks towards malicious prompt detection, in: 2024 5th International Conference on Electronics and Sustainable Communication Systems (ICESC), IEEE, 2024, pp. 1531–1535.

[27] S. Sukhbaatar, E. Grave, P. Bojanowski, A. Joulin, Adaptive attention span in transformers, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, 2019, pp. 331–335.

[28] K. Giapantzis, S. T. Halkidis, Xlcnn: A transformer model for malware detection, International Journal of Computer Science and Information Security (IJCSIS) (2023).

[29] E. B. Wilson, Probable inference, the law of succession, and statistical inference, Journal of the American Statistical Association 22 (1927) 209–212.

[30] P. Chao, E. Debenedetti, A. Robey, M. Andriushchenko, F. Croce, V. Sehwag, E. Dobriban, N. Flammarion, G. J. Pappas, F. Tramer, et al., Jailbreakbench: An open robustness benchmark for jailbreaking large language models, Advances in Neural Information Processing Systems 37 (2024) 55005–55029.

[31] T. Xie, X. Qi, Y. Zeng, Y. Huang, U. M. Sehwag, K. Huang, L. He, B. Wei, D. Li, Y. Sheng, et al., Sorry-bench: Systematically evaluating large language model safety refusal, arXiv preprint arXiv:2406.14598 (2024).