information

MDPI

Article

Comparative Benchmarking of Deep Learning Architectures for Detecting Adversarial Attacks on Large Language Models

Oleksandr Kushnerov [1], Ruslan Shevchuk [2,3,*], Serhii Yevseiev [4] and Mikołaj Karpiński [5,6]

[1] Department of Economic Cybernetics, Academic and Research Institute of Business, Economics and Management, Sumy State University, Kharkivska Str., 116, 40007 Sumy, Ukraine; o.kushnerov@biem.sumdu.edu.ua
[2] Department of Computer Science and Automatics, University of Bielsko-Biala, Willowa Str., 2, 43-309 Bielsko-Biala, Poland
[3] Department of Computer Science, West Ukrainian National University, Lvivska Str., 11, 46009 Ternopil, Ukraine
[4] Department of Cyber Security, Educational and Scientific Institute of Computer Science and Information Technologies, National Technical University "Kharkiv Polytechnic Institute", Kyrpychova Str., 2, 61002 Kharkiv, Ukraine
[5] Department of Software Engineering, Institute of Security and Computer Science, University of the National Education Commission, Podchorążych Str., 2, 30-084 Krakow, Poland; mikolaj.karpinski@uken.krakow.pl
[6] Department of Cybersecurity, Ternopil Ivan Puluj National Technical University, Ruska Str., 56, 46001 Ternopil, Ukraine
* Correspondence: rshevchuk@ubb.edu.pl

Abstract

The rapid adoption of large language models (LLMs) in corporate and governmental systems has raised critical security concerns, particularly prompt injection attacks exploiting LLMs' inability to differentiate control instructions from untrusted user inputs. This study systematically benchmarks neural network architectures for malicious prompt detection, emphasizing robustness against character-level adversarial perturbations—an aspect that remains comparatively underemphasized in the specific context of prompt-injection detection despite its established significance in general adversarial NLP. Using the Malicious Prompt Detection Dataset (MPDD) containing 39,234 labeled instances, eight architectures—Dense DNN, CNN, BiLSTM, BiGRU, Transformer, ResNet, and character-level variants of CNN and BiLSTM—were evaluated based on standard performance metrics (accuracy, F1-score, and AUC-ROC), adversarial robustness coefficients against spacing and homoglyph perturbations, and inference latency. Results indicate that the word-level 3_Word_BiLSTM achieved the highest performance on clean samples (accuracy = 0.9681, F1 = 0.9681), whereas the Transformer exhibited lower accuracy (0.9190) and significant vulnerability to spacing attacks (adversarial robustness $\rho_{spacing} = 0.61$). Conversely, the Character-level BiLSTM demonstrated superior resilience ($\rho_{spacing} = 1.0$, $\rho_{homoglyph} = 0.98$), maintaining high accuracy (0.9599) and generalization on external datasets with only 2–4% performance decay. These findings highlight that character-level representations provide intrinsic robustness against obfuscation attacks, suggesting Char_BiLSTM as a reliable component in defense-in-depth strategies for LLM-integrated systems.

Keywords: language models; prompt injection; adversarial machine learning; neural networks; symbolic vectorisation; model robustness; attack detection; artificial intelligence cybersecurity

Check for updates

Academic Editors: I-Pin Chang and Tian-Fu Lee

Received: 30 December 2025
Revised: 28 January 2026
Accepted: 2 February 2026
Published: 4 February 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

The deployment of large language models (LLMs), including GPT-4, Claude, and Llama, has introduced new security risks that conventional information protection frameworks struggle to mitigate, particularly when these models interact with external systems, Retrieval-Augmented Generation (RAG), and multi-turn conversational environments [1,2]. The rapid expansion of LLM applications, from financial decision support to code generation, has outpaced the development of specialized security protocols, creating a persistent vulnerability in critical infrastructure [3–5]. The absence of tailored protective measures has rendered these systems susceptible to cyberattacks. Among these threats, prompt injection attacks (PIAs) have emerged as particularly severe, as adversaries can manipulate input queries to execute system commands, exfiltrate protected information, and perform unauthorized operations [1,6,7].

We formalize a threat model in which an adversary has full control over the input string, aiming to compromise the unified token stream where instructions and user data coexist. By exploiting these manipulations, attackers can override system prompts, access hidden information, and trigger tool executions without authorization, reflecting LLMs' limitations in distinguishing control commands from standard input [7–10]. Detection of PIAs requires context-aware semantic analysis, as these attacks differ fundamentally from traditional injections such as SQL injection.

Production environments, including LLM-augmented customer support, code generation pipelines, and document processing systems, are particularly vulnerable to PIAs that initiate unauthorized API requests, extract confidential data, and manipulate system responses [5,11]. Moreover, the attack surface extends beyond direct user prompts, as malicious instructions can be embedded in external documents or web content retrieved by the model, resulting in indirect PIAs [7].

Adversarial prompt injection threats not only pose a risk to complex intelligent systems regarding the bypassing of control frameworks (jailbreaking) [12,13], but also carry risks regarding privacy violations, including the exposure of personally identifiable information (PII), unauthorized extraction of model parameters, and poisoning of trusted knowledge bases designed for RAG-based models [2,14]. Furthermore, in scenarios where LLM agents function as natural language middleware for sensitive operations—such as drafting parameters for Certificate Signing Requests (CSRs) in Public Key Infrastructure (PKI) or interpreting policies for key management systems (KMSs)—PIAs introduce a novel attack surface. In this proposed bounded threat model, the detection mechanism acts as a pre-computation sanitization layer. The adversarial input targets the LLM's interpretation logic rather than the underlying cryptographic primitives, potentially inducing the system to process unauthorized transactions under the guise of valid user intent [5,7]. Authoritative taxonomies, including NIST AI 100-2e2023 and OWASP Top 10 for LLMs, classify prompt injection as a distinct adversarial threat requiring specialized detection mechanisms beyond traditional content moderation [6].

Prompt injection detection. The task of identifying malicious prompts is formalized as a binary sequence classification problem. Given $X = \{x_1, x_2, \ldots, x_n\}$ as an input sequence of either tokens or characters, the goal is to minimize the loss function for the function $f(X) \rightarrow \{0, 1\}$, where y = 1 indicates malicious manipulation:

$$\min_{\theta} \mathscr{L}(f(X;\theta), y)$$

Adversaries employ obfuscation techniques, including homoglyph substitution, Unicode manipulation, and whitespace injection, to evade word-level tokenizers while preserving semantic malicious intent. Although advanced encoding methods (Base64 and

ROT13) and structural manipulations (context reversal) are also observed in real-world attacks, this study focuses on character-level perturbations as a foundational robustness benchmark, with broader obfuscation strategies acknowledged as important directions for future evaluation [5,8,15].

The primary contribution of this research is a comparative adversarial benchmarking of eight deep learning architectures (Dense DNN, CNN, BiLSTM, BiGRU, Transformer, ResNet, Character-level CNN, and Character-level BiLSTM), focused on detection accuracy and resilience against character-level obfuscation within a hold-out validation framework. Existing multi-layered defense frameworks, such as GUARDIAN [16] and PSF [17], implement tiered filtering but generally lack mechanisms to counter character-level perturbations and often neglect the trade-off between multiple filtering stages and inference latency in real-time applications. This study addresses these limitations by evaluating how neural and symbolic architectures can serve as robust, low-latency primary filters within a layered defense-in-depth strategy. It provides a theoretical foundation for selecting optimal architectures that balance high resistance to character-level attacks with minimal inference overhead, offering practical guidance for secure deployment in LLM-integrated systems.

The importance of this work is highlighted by the need for efficient classification and short processing time. Experimental outcomes clearly reveal that RNN-based models, specifically 3_Word_BiLSTM, achieve significant validation accuracy (Acc = 0.9681, AUC = 0.9918), but with slightly higher latency (≈0.31 ms) in comparison to models with convolutional layers. Furthermore, this research demonstrates that character-level models, specifically 8_Char_BiLSTM, provide significant advantages against adversarial attacks such as Malignant, achieving a detection accuracy of 0.5471. This is approximately 2.5 times higher than word-level counterparts, which, from a representation learning perspective, lack the necessary sequence inductive bias to maintain semantic integrity when token patterns are disrupted by noise. Such vulnerability suggests that word-level models may be insufficient as standalone filters for interfaces requiring high-assurance input validation, particularly where symbolic obfuscation is a primary evasion tactic. The importance of these findings supports an adaptive defense strategy for security architecture design, balancing high-performance CNN architectures (latency ≈ 0.03–0.10 ms) with specialized layers focused on achieving maximum resistance against obfuscated injections. This approach is critical for minimizing the attack surface for sensitive downstream components, such as those found in PKI-enabled workflows, where LLMs act exclusively as semantic access interfaces rather than direct cryptographic operators or security enforcement points. These findings provide a path forward for security teams to integrate lightweight, robust detectors into CI/CD pipelines and production environments, enabling a "cost-risk" approach to LLM deployment. By selecting architectures based on their specific resilience to obfuscation, organizations can enforce stricter security policies without compromising system performance.

2. Related Work

On the basis of the existing research environment regarding the security of LLMs, it is observed that the development of this multi-level threat environment has taken the form of detecting adversarial attacks not only from the perspective of a technological problem but, rather, an essential component within the protection of the confidentiality and integrity of intelligent systems. The primary categorization criterion for such threats is the National Institute of Standards and Technology (NIST) Taxonomy, as specified in AI 100-2e2023 [6]. According to the NIST, Adversarial Machine Learning (AML) attacks are defined by their potential to breach the "three pillars of information security—integrity, availability, and confidentiality". In this respect, prompt injection and indirect prompt injection (sometimes

referred to as malicious cues) have been identified as specific influence vectors that exploit the semantic processing of transformer-based architectures to manipulate the model's target task. These categories, specified by the NIST as "evasion attacks", indicate that the attacker does not change the "model parameters" but coerces the model to create an "unintentional, harmful" output. This aligns with the OWASP LLM01:2025 classification [18], where PIAs are prioritized as a critical threat due to the inherent inability of LLMs to distinguish between developer instructions and user-provided input.

The economic and applied implications of these vulnerabilities are analyzed in depth by Kang et al. [4]. In particular, the authors illustrate how strong translation capabilities make modern LLMs the greatest attack tool by excellence. In fact, the study proves how injecting the creation of malicious content (phishing and scams) is between 125 and 500 times cheaper than manual labor, prompting cybercriminals to commit attacks. In the case of specific tasks, like machine translation, the work of Miceli Barone and Sun et al. [8] presents an extensive testbed using the WMT 2024 test suite, demonstrating how even constrained tasks remain fragile to "ignore translation instructions" attacks. Such attacks, performed in JSON/1-shot format, prove to be the most successful, forcing the model to output the underlying system prompt or even to answer hidden questions rather than translating.

Considerable efforts in the research community have focused on integrated LLM applications in which models utilize external tools like web search functionality, APIs, and knowledge repositories. The foundational work by Greshake et al. [7] revealed the mechanisms of indirect injections, where an attacker can control model behavior by poisoning data retrieved from online repositories. For example, an attacking instruction posted on a website can trigger the LLM email agent to send confidential emails to the attacker's server, thus posing serious risks to the confidentiality of systems that process confidential data in real time online. This is supported by practical analyses, such as those by Kantor [9], which demonstrate that online attack behaviors employ obfuscation mechanisms (e.g., payload splitting and virtualization) that render identification via keyword searches ineffective.

The role of privacy and protection of confidential information (PII) within LLM-based agents is also a prominent issue within current cryptographic studies. Yan et al. [2], within their systematic reviews on High-Confidence Computing, categorize threats as follows: those occurring via passive leaks (accidental discovery due to training data leakage) and those originating from active attacks (targeted information extraction). These researchers pointed out that the use of big models often results in the "memorization" of private sequences, which allows an attacker to recover names, addresses, and private keys through the use of specifically designed PIAs. Regarding Retrieval-Augmented Generation (RAG), the SafetyRAG framework [14] establishes a trust boundary by implementing "security facts" to neutralize malicious information. This approach aligns with broader trends in trustworthy AI, where the retrieval process is fortified against poisoning attacks. Specifically, Omri et al. [14] demonstrate that integrating a "security facts" model enables the efficient removal of malicious content without the computationally demanding process of model retraining, thereby preserving the integrity of the knowledge base. Once again, data integrity is considered a relevant aspect within the scope of published scientific results, as shown within a study conducted by Rao et al. [19], which addressed the use of LLM-generated peer reviews and highlighted vulnerabilities related to indirect injections aimed at manipulating academic rankings.

The current design methods for addressing PIAs in architecture are generally considered to follow built-in alignment methods and external barriers. However, the method of Red Teaming, explained in the report authored by Trabilsy et al. [3], referring to the PIEE cycle, and Milani et al. [20], relating to educational risks, shows that "aligned" models can easily succumb to semantic manipulations. As for the above-mentioned limitations, exter-

nal architectures such as GUARDIAN [16] and the Prompt-Shield Framework (PSF) [17] have been suggested to address the shortcoming. The GUARDIAN system employs a three-tier architecture consisting of fast pre-filtering, deep neural toxicity classification, and output verification. However, similar to the PSF, it remains primarily focused on word-level semantics, leaving a research gap in defending against character level adversarial perturbations. In the report authored by Hadiprakoso et al. [17], the PSF framework, the Llama 3.2 local model and the modules "CAP," "OV," and "SFL" are introduced for real-time adaptive filtering, achieving 97.83% accuracy.

Similarly to the "smart guardrails" methods proposed by Shvetsova et al. [12], there is a growing consensus that security should be integrated directly into the CI/CD pipelines of LLMs. Furthermore, specialized LLM fine-tuning for security tasks, analogous to established paradigms in spam filtering, provides a precedent for using custom architectures to enhance classification performance.

Recent advances in trustworthy detection emphasize the analysis of the model's internal states and explainability (XAI) as essential components for enforcing security policies. Attention Tracker, proposed by Hung et al. [10], is based on the detection of a distraction effect. The authors proved that in a successful injection, a certain neural head suddenly changes its attention from the original to the attack command. The training-free technique resulted in a 10% relative improvement in AUROC and shows promise for lightweight detectors inside the transformer framework. Concurrently, a Fine-Grained Chain-of-Thought analysis was proposed by Shi et al. in their Meticulous Thought Defender [15] to audit the model's reasoning logic, identifying attacks through semantic inconsistency at each generation stage. The process of vectorization and optimal classifier choice is still an area of debate. A comparative analysis of NLP techniques (TF-IDF, Word2Vec, and BERT) by Jain et al. [21] indicates that recurrent neural networks (RNNs) are more effective at modeling sequential patterns in misleading examples. Ergün and Onan in [22] introduced a cascading CNN-LSTM model, integrating localized feature identification and analysis of global patterns, reporting accuracy above 97%. However, the above studies mostly focus on normalized text at the word level. While foundational adversarial NLP literature has long explored character-level attacks and robust modeling, their specific application to prompt injection detection remains significantly underaddressed compared to word-level semantic filters. In their IJNDI analysis, Lan et al. [11] use both similarity search and BERT classifier for a fast detection solution but point out the susceptibility of proposed models to Unicode attacks. Explainability (XAI) in the context of cybersecurity solutions is viewed as a tool for strengthening confidence among detectors. An approach to developing solutions was demonstrated by Sayeedi et al. [13] through JailbreakTracer, which uses synthetic data generation and explainability AI to graphically represent the triggers of attacks in text. This is consistent with the approach described by Mirtaheri et al. [1], which uses a systematic risk management taxonomy and considers XAI to be a fundamental component of auditing for GenAI systems. The quality of data is considered a problem highlighted in [23], which claimsthat most existing datasets for PI tend to be too simple. The utilization of the MPDD [24] in this work attempts to present more complicated examples of attacks. From a cryptography point of view, our work is anchored in the existing literature regarding the security of KMS and PKI. As noted by Balakrishnan and Leema [5], LLMs increasingly serve as natural language interfaces to cryptographic services. In this context, prompt injection represents a logical attack vector, where the model could be manipulated to initiate unauthorized signing requests or misinterpret policy constraints, thereby bypassing semantic validation layers. Our findings regarding ResNet's vulnerability (robustness score of 0.02) compared to Char-BiLSTM's resilience (0.98) suggest that character-level (symbolic) vectorization provides a critical defense-in-depth component for designing secure,

policy-aware interfaces in cryptographic infrastructures. Despite the considerable amount of work, the current academic discourse highlights several crucial research gaps. Firstly, despite various studies on individual models, there is a lack of systematic comparative evaluations across heterogeneous neural architectures (CNN, BiLSTM, Transformer, and ResNet) specifically focused on the trade-off between adversarial robustness and inference latency for real-time pre-filtering [16,21,22]. Secondly, the problem of underestimating the robustness of detectors concerning adversarial perturbations on the symbolic level (spacing, homoglyphs) arises, in which case word-wise models might be utterly inadequate. Thirdly, the literature barely touches upon the trade-off between accuracy and computational complexity, which becomes a crucial aspect when real-time systems are involved. This work addresses these gaps by providing a decision-ready benchmark of eight architectures, serving as a framework for building future-ready, policy-aware defense layers that match the pace of modern LLM security challenges.

3. Materials and Methods

The methodological concept of this study is predicated on a multi-layered security paradigm, evaluating deep learning architectures to identify the optimal "cost-risk" configuration for prompt injection detection. The primary objective is to establish a robust detection pipeline capable of identifying Adversarial Prompt Injection under conditions of symbolic obfuscation designed to circumvent standard token-based security systems. The experimental design adheres to transparency standards in artificial intelligence and cybersecurity to ensure reproducibility. The present study focuses on meeting NIST taxonomic requirements to maintain generative models' stability and confidentiality, as outlined in references [1,5,6].

The research process starts with the essential step of building an experimental foundation, which requires the initial task of analyzing and confirming a representative dataset. In this study, a specialized corpus was selected for analysis. The study utilizes the MPDD [24], comprising 39,234 instances. After automated filtering and expert inter-annotator agreement verification (Cohen's Kappa = 0.9), the final corpus was balanced with 19,617 (50%) malicious and 19,617 (50%) benign prompts. The dataset allows researchers to develop models which detect multiple threat types, including both direct attempts to bypass ethical filters through jailbreak, and indirect injection attacks that manipulate autonomous agent contexts during extended generation systems (RAGs) [7,14]. The main feature that sets MPDD apart is its diverse nature, which stretches beyond direct harmful instructions to include hidden language patterns that emerge from the transformation of system architectures. The dataset contains multiple role-playing exercises, which force the model to ignore system commands because of simulated situations. The system provides quick output methods that hide malicious commands within translation or generalization requests.

The methodology for developing the final sample was predicated on a three-tiered inspection protocol that serves to minimize labeling entropy while ensuring maximum objectivity. The initial phase required the execution of automatic filtering by using heuristic algorithms which successfully eliminated syntactically incorrect fragments. System logs, blank lines, and code fragments served no semantic purpose and were considered noise. The subsequent stage entailed preliminary classification via the implementation of sophisticated language models that functioned as expert judges, thereby unveiling latent malicious intentions. The automated system identified the most complex examples during this stage because malicious intentions stay hidden in single words but emerge through the overall context of the instructions. The third phase required a team of cybersecurity and linguistics specialists to perform a consensus verification. The samples underwent strict evaluation procedures to assess their adversarial nature and their ability to meet attack

success standards based on the established relevance and danger scale. The high quality of the labeling was confirmed by Cohen's Kappa coefficient (Kappa = 0.9), indicating almost complete agreement among experts and allowing this array to be used as the gold standard for neural network detector training [20]. This study focuses on the detection of malicious prompts through binary classification, treating successful attack prevention as an indirect privacy safeguard. While we analyze the effectiveness of detection architectures in blocking PIAs that could lead to PII exfiltration or unauthorized data access, we do not directly measure privacy-specific metrics such as leakage rate, PII extraction success rate, or tool-use exfiltration probability under active adversarial conditions. Direct privacy guarantees require additional mechanisms beyond detection, including differential privacy, input sanitization pipelines, and policy-aware access control, which fall outside the scope of this comparative architectural study.

To ensure experimental rigor and prevent data leakage, the dataset was partitioned using a stratified hold-out validation split into three disjoint subsets: training (70%), validation (15%), and testing (15%). Crucially, all hyperparameter tuning and architectural optimization (including grid search) were performed exclusively using the training and validation sets. The testing set was strictly held out and utilized only for the final performance evaluation, ensuring that reported metrics reflect the models' true generalization capability on unseen data [15,21]. The indicators' priority was shifted towards resistance to type II errors (false negative results), but special attention was also paid to minimizing the frequency of false positive results (FPRs). In intelligent systems, erroneous blocking of a legitimate user request not only critically reduces usability but also places an additional burden on the support system and may lead to the disclosure of the very fact of the functioning of protective mechanisms, which is contrary to the principles of data privacy and privacy systematized in fundamental scientific reviews [2].

As illustrated in Figure 1, the methodological structure of the study is founded on the principle of sequential filtering and multi-vector verification. The architectural logic of the pipeline entails a fundamental transformation of the input request through a layer of full-text normalization. At this juncture, invisible control characters are eliminated, spaces are standardized, and HTML tags that can be utilized to mask attacks are cleared. A fundamental aspect of the approach involves the further bifurcation into two distinct vectorization branches. This facilitates parallel analysis of semantic features at the level of high-level token representation and concurrent regulation of structural anomalies at the level of individual symbols. The visualization corroborates the hypothesis that each of the eight neural network topologies studied undergoes a single stage of adversarial evaluation. The experimental design under consideration is intended to minimize the influence of random factors, thereby highlighting the net increase in the Ro gap provided by the internal architectural features of each model. This approach is intended to prioritize the impact of these architectural features rather than the quality of the input characteristics. To quantify resistance to adversarial noise, we define the robustness coefficient ρ as the ratio of the F1-score on a perturbed dataset to the performance on the clean test set $\left(F_1^{clean}\right)$:

$$\rho = \frac{\mathrm{F}_1^{\mathrm{adv}}}{\mathrm{F}_1^{\mathrm{clean}}}$$

where F_1^{adv} represents the model's harmonic mean under specific character-level attacks (spacing or homoglyphs). A value of $\rho \approx 1.0$ indicates absolute immunity to the specific attack type.

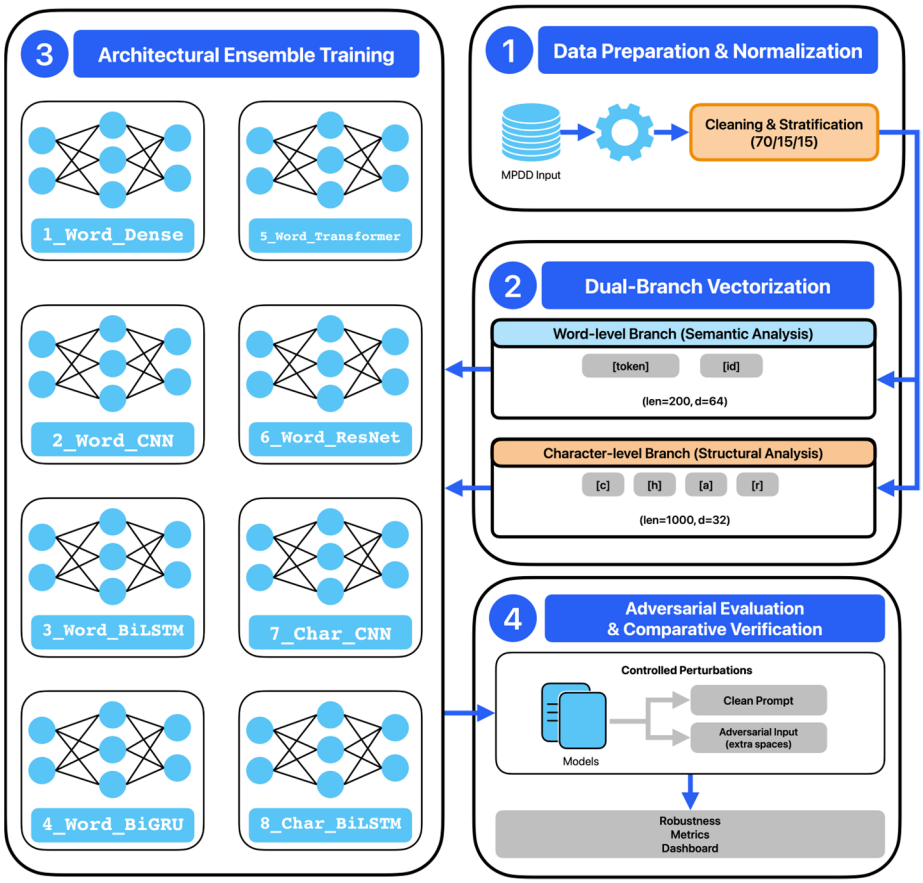

Figure 1. Conceptual scheme of the experimental pipeline and procedures for comparative analysis of detection architectures.

Given that LLM attacks frequently exploit vulnerabilities in tokenization processes (e.g., breaking tokens through the use of special characters, replacing homographic characters, or encoding), a dual vectorization strategy is implemented that considers text at varying levels of abstraction. Word-level vectorization is predicated on the extended dictionary $\mathcal{V}_w$ of the 20,000 most frequent tokens collected from the training set. Each input sequence undergoes a rigorous normalization stage, which includes conversion to lowercase, removal of extra spaces, and filtering of non-standard Unicode characters. Subsequent to the normalization process, the text is converted into a fixed-length index vector $L_w = 200$. This length was determined through a meticulous statistical examination of MPDD, which revealed that the overwhelming majority (over 95%) of malicious instructions are limited to 180 tokens. In the event that this limit was exceeded, a truncation technique was employed, as it has been empirically demonstrated that critical injection signatures are typically concentrated in the prefix part of the query. This phenomenon can be attributed to the attacker's objective of disrupting the system role of the language model at the inception of the context window. For the purpose of semantic representation, trained embeddings with dimensions of $d_w = 64$ are utilized, which are optimized directly during the classification task training.

Concurrent with the representation of the token based on the alphabetical sequence $\mathcal{V}_c$, a representation was formulated at the character level for each sequence with length $L_c = 1000$. This alphabet comprises 100 fundamental characters. The characters include Latin, Cyrillic, numbers, mathematical operators, and common special characters. Character-level vectorization is essential for this study, as it enables sensitivity to low-level manipulations ignored by dictionary tokenizers. Homoglyphic substitution attacks exploit this by substituting a Latin character with a visually similar character from another script

(e.g., Latin 'a' vs. Cyrillic 'a'). These manipulations do not affect human readability; rather, they generate entirely new tokens for the model, thereby circumventing filters. Therefore, the employment of character branches constitutes a fundamental condition for the countering of attacks such as Homoglyph and Spacing. In these attacks, an attacker attempts to render a malicious word "invisible" to a standard analyzer while preserving its malicious meaning [3,8,10].

The mathematical formalization of the architectures under study encompasses eight heterogeneous neural network topologies. The initial group of models is predicated on word vectorization and is conceived to apprehend the semantic substance of instructions. For the fundamental Dense DNN model (1_Word_Dense), the output is determined by successive nonlinear transformations of the feature vector x, which is formed by global averaging of inputs along the time axis. Mathematically, this phenomenon can be expressed as follows:

$$y = \sigma(W_n(\ldots ReLU(W_{1x} + b_1)\ldots) + b_n)$$

where W_i and b_i are the weight matrices and offset vectors of the network layers, respectively, and σ is the logistic sigmoid activation function. This model serves as a reliable baseline for evaluating the advantages of more complex architectures.

The second architecture, CNN (2_Word_CNN), uses layers of one-dimensional convolutional filters to identify local n-gram patterns that are characteristic signatures of attacks. For each filter with weights w and width k, the activation is calculated:

$$x_i^{conv} = ReLU\left(\sum_{j=0}^{k-1} w_j \cdot x_{i+j} + b\right)$$

After that, a global maximum union operation is applied, which records the presence of key tokens regardless of their exact position in the text. This mechanism allows the model to recognize specific phrases or code fragments that are often found in malicious prompts. Experiments have shown that convolutional models have the best learning speed and ultra-low inference latency, making them ideal candidates for deployment in high-throughput real-time systems [7].

The third and fourth models belong to the class of recurrent networks—BiLSTM (3_Word_BiLSTM) and BiGRU (4_Word_BiGRU). Their mathematical apparatus is based on a gate mechanism that regulates the flow of information through hidden states, allowing the modeling of long-term contextual dependencies. In the LSTM model, the state of cell c_t at each step t is defined as

$$c_t = f_t \odot c_{t-1} + i_t \odot \tanh(W_c[h_{t-1}, x_t] + b_c)$$

where f_t is the forget valve, i_t is the input valve, and $\odot$ denotes elementary multiplication. The use of bidirectional blocks allows the model to analyze the query simultaneously in forward and reverse time directions, which significantly improves the detection of attacks when malicious instructions are scattered throughout the text or hidden at the very end of a long prompt [13,17]. The fifth model is based on the Transformer architecture (5_Word_Transformer), where the Multi-head Self-Attention mechanism plays a key role:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V$$

However, empirical results indicate that transformer-based architectures exhibit significant susceptibility to character-level noise. This vulnerability is primarily attributed to a lack of sequence inductive bias within the self-attention mechanism, which leads to "at-

tention dispersion". Under adversarial perturbations, the attention heads fail to effectively aggregate global semantic features from fragmented or non-standard tokens, resulting in a collapse of the model's decision-making logic. The sixth model, ResNet (6_Word_ResNet), uses skip connections to overcome the gradient vanishing problem:

$$y = \mathcal{F}(x, \{W_i\}) + x$$

This allows for effective training of deep hierarchical representations of text data. The second group of models focuses on character vectorization: Character-level CNN (7_Char_CNN) and Character-level BiLSTM (8_Char_BiLSTM). Unlike word models, these architectures work directly with character embeddings, $d_c = 32$. This enables the system to autonomously discern structural anomalies in garbled text at the level of individual letters and characters, independent of the stability of the token dictionary. This is a critical factor in ensuring privacy, as attackers often attempt to extract personal data via requests encoded using special methods (such as Base64 or the use of custom fonts) that completely destroy word structure but remain transparent to character analysis [3,11].

The mathematical optimization of the models was achieved by minimizing the binary cross-entropy loss function:

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log(\widehat{y_i}) + (1 - y_i)\log(1 - \widehat{y_i})]$$

The minimization of this feature enables the model to maximize confidence in the accuracy of the classification, while concomitantly penalizing erroneous predictions. The weights were updated using Adam's algorithm, a sophisticated method that integrates the benefits of two other algorithms, namely, adaptive gradient descent (RMSProp) and momentum (Momentum). The update of the θ parameters is achieved through the application of the following equations:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2$$

$$\widehat{m_t} = \frac{m_t}{1 - \beta_1^t}, \ \widehat{v_t} = \frac{v_t}{1 - \beta_2^t}$$

$$\theta_t = \theta_{t-1} - \frac{\alpha}{\sqrt{\widehat{v_t}} + \epsilon}\widehat{m_t}$$

In this equation, g_t denotes the gradient of the loss function, α is the learning speed, and β_1 and β_2 represent the hyperparameters of moments.

A particular emphasis was placed on the development of an embedded layer for both levels of vectorization. Mathematically, this layer is represented by a matrix E belonging to the set of real matrices $E \in R^{V \times d}$. For CNN's character model, three and five convolution kernels were utilized at the character level, enabling the network to concurrently capture both short and long character sequences. This facilitated the identification of anomalies in the morphological structure of the clues.

A critical stage of the methodology is the simulation of an adversarial attack. The study incorporated two categories of perturbations. We simulate two primary adversarial vectors:

Spacing Attack: A transformation $T_{adv} = \bigcup_{i=1}^{n}\{c_i, ' '\}$

1. that forces word-level tokenizers into Out-of-Vocabulary (OOV) states.
2. Homoglyph Attack: Latin characters (e.g., 'a' U+0061) are replaced with visually identical Unicode glyphs (e.g., Cyrillic 'a' U+0430) to bypass encoding-based filters. While we acknowledge advanced obfuscations like Base64 encoding or ROT13, this study focuses on these character-level perturbations as foundational robustness benchmarks.

From a mathematical perspective, an adversarial attack can be conceptualized as the task of identifying a minimum disturbance, denoted by the symbol δ. This results in an incorrect classification:

$$\min|\delta| \text{ s.t. } f(X+\delta) \neq y$$

The subsequent equation formalizes the adversary's strategic objective. The objective is to ascertain the minimum perturbation that remains imperceptible to conventional monitoring systems (e.g., filters based on regular expressions or keywords) while concurrently effecting a substantial alteration in the logical outcome of the neural network classifier. Simulating threats against sensitive interfaces (e.g., KMS and PKI) necessitates modeling an adversary delta in the form of character substitutions, which preserve the visual appearance of administrator request messages. The tokenizer dictionary, which was trained through word models, fails to detect malicious commands that these substitutions can use. The training requires hyperparameter adjustments to minimize interference effects. The models will maintain their resolution stability through this method when they encounter conflict noise. The optimized learning settings are delineated in Table 1.

Table 1. Hyperparameters and settings of the training process for injection detection models.

Parameter	Value	Description and Justification
Optimizer	Adam	Adaptive moment estimation algorithm to ensure stable convergence of gradients
Learning rate (lr)	10^{-3}	Optimal learning rate determined based on preliminary grid search analysis on the validation set
Loss function	Binary Cross-Entropy	Standard objective function for binary classification tasks of adversarial attacks
Batch size	128	Strikes a balance between stable weight update and efficient GPU memory usage
Regularization	Dropout (0.3–0.5)	Used to prevent overtraining of models on specific tokens
Early stopping	Patience = 3	Mechanism to stop learning when val_loss does not improve
Initialization of weights	Glorot Uniform	Provides uniform dispersion of activations at the initial stages of training

Table 1 shows how the parameter selection process was carefully designed to improve the models' generalization capabilities. Latency benchmarks were executed on an Apple M4 Max SoC (Apple Inc., Cupertino, CA, USA) (16-core CPU, 40-core GPU, 36GB RAM) using the TensorFlow Metal (MPS) backend (Apple Inc., Cupertino, CA, USA). To ensure statistical rigor, we report the mean inference time for a batch size of 1, calculated over 1000 runs following a 100-run warm-up period to eliminate "cold start" bias and thermal throttling effects. The Adam optimizer helped the model reach stable gradient convergence, which serves as a fundamental requirement for training deep recurrent networks. The network started to depend on more stable semantic features, which did not require particular word markers because sifting layers operated at 0.5 probability levels. The model's ability to detect new attacks remained intact because the early stop mechanism eliminated the requirement for retraining. The experimental pipeline was fully implemented in Python (version 3.13, Python Software Foundation, Wilmington, DE, USA), and the TensorFlow framework (version 2.20.0, Google LLC., Mountain View, CA, USA) and Scikit-learn library (version 1.8.0, NumFOCUS, Austin, TX, USA) were used to pre-process and evaluate metrics.

A significant component of the methodological framework is the procedure for assessing the generalizability of external datasets that have not been incorporated within the body of the MPDD. To evaluate cross-domain generalization, we utilized five external datasets: Jailbreaks (3456 samples), Forbidden Qs (2100 samples), Malignant (1500 highly obfuscated prompts), Safe Quora (5000 benign queries), and Prediction Guard (800 prompt leakage attempts). None of these samples overlapped with the MPDD training set. The evaluation process allowed researchers to test how well the detectors could identify attacks which used different language patterns and distinct manipulation strategies. To this end, a cross-domain testing protocol was developed in which models trained on the same type of attack were evaluated in fundamentally different scenarios. This approach facilitates the identification of architectures that understand the underlying structure of adversarial influence at the semantic level.

The final stage of the methodology entailed a "human-in-the-loop" verification procedure to ensure that adversarial examples maintain their original malicious meaning to humans. The core element of the study demands this component to achieve its complete validity. The detection process turns into a basic noise detection task when the attack makes the text completely unreadable. A team of independent experts analyzed a selection of generated perturbations to confirm that human perception fails to detect the concealed malicious content, even though the symbols used are technically complex. All experiments needed to maintain a fixed random number generator value to achieve statistical reliability in the results. The random number generator was operated with the Random Seed = 42 configuration. The evaluation protocol used standard training and testing sample division for evaluation yet performed detailed performance assessments through confusion matrix analysis of each architecture. The research team discovered which types of attacks create the most problems for word models. The validation set was utilized to optimize the architectural parameters, which included CNN filter count and BiLSTM block hidden layer dimensions. Together, these methods create a strong base, which enables the creation of security systems for LLMs. The research results suggest that symbolic architectures can serve as a foundational component for protecting data privacy when defending against cyber threats, acting as a robust filtration layer against evasion attacks. These keep evolving [6–8].

4. Results

We conducted a comparative evaluation of eight static deep learning architectures to establish a performance benchmark for detecting PIAs under varying levels of symbolic noise. This section presents a comprehensive assessment of modern neural network topologies to determine their efficacy in shielding LLM systems from security breaches and privacy violations. The testing protocol comprised three essential components: first, evaluating baseline classification performance on clean data to establish false positive rates [24]; second, estimating the computational complexity of inference; and third, conducting extensive stress testing against adversarial attacks. To ensure reproducibility, all computational experiments were performed on an Apple MacBook Pro workstation equipped with an Apple Silicon SoC and 36 GB of Unified Memory, running macOS Sequoia 15.1. The inference latency was averaged over 1000 runs following a 100-run warm-up phase to eliminate JIT-compilation overhead and ensure statistical stability.

All models were evaluated on the same test sample to achieve scientific validity. The dataset was partitioned using a stratified split strategy to maintain class balance (50% benign, 50% malicious). The quantitative distribution of samples across the training, validation, and testing subsets follows a 70:15:15 ratio, as detailed in Table 2.

Table 2. Dataset distribution and split statistics.

Subset	Percentage	Samples (Total)	Class Distribution (Benign/Malicious)	Purpose
Training	70%	27,462	13,731/13,731	Model parameter optimization
Validation	15%	5886	2943/2943	Hyperparameter tuning and early stopping
Testing	15%	5886	2943/2943	Final performance evaluation (Clean and Adversarial)

The mathematical apparatus for evaluation was based on minimizing the binary cross-entropy function:

$$\mathscr{L} = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log(\widehat{y_i}) + (1 - y_i)\log(1 - \widehat{y_i})]$$

This enabled models to enhance their predictive accuracy for malicious intent while reducing false positives, a crucial aspect for ensuring the efficacy of privacy protection systems. Preliminary examination of accuracy, harmonic mean (F1), and Area Under the Curve (AUC) highlights the constraints inherent in each architecture under optimal operating conditions. The empirical performance of the evaluated architectures on the clean test set, along with complexity benchmarks, is summarized in Table 3.

Table 3. Comparative characteristics of the detection power and complexity of the models.

Model	Accuracy	AUC ROC	Latency (ms)	Robustness (adv)	Params
1_Word_Dense	0.9551	0.9828	0.0289	0.524	1,282,113
2_Word_CNN	0.9621	0.9898	0.0425	0.556	1,302,657
3_Word_BiLSTM	0.9681	0.9918	0.3088	0.482	1,354,369
4_Word_BiGRU	0.9613	0.9909	0.2506	0.494	1,338,241
5_Word_Transformer	0.9190	0.9190	0.3078	0.508	1,301,153
6_Word_ResNet	0.9630	0.9905	0.0592	0.536	1,308,929
7_Char_CNN	0.9631	0.9872	0.0993	0.874	23,873
8_Char_BiLSTM	0.9599	0.9803	0.7757	0.704	21,953

A thorough examination of the data presented in Table 3 unveils subtle variations in the efficacy of architectures in detecting clean data attacks. Specifically, the 3_Word_BiLSTM model demonstrated the highest level of accuracy (accuracy = 0.9681) and the largest Area Under The Curve (AUC = 0.9918), thereby substantiating the hypothesis that bidirectional memory plays a pivotal role in recognizing complex instructions where malicious content can be disseminated between the query's inception and conclusion. The recurrent structure enabled the model to effectively disregard the neutral semantic noise of legitimate queries, enabling it to prioritize the sequence of instructions. Conversely, the 5_Word_Transformer architecture exhibited the lowest baseline performance (accuracy = 0.9190; AUC = 0.9190). From a representation learning perspective, this is explained by the Transformer's lack of inherent sequence inductive bias; unlike RNNs, it relies entirely on positional encodings which, in the context of relatively small and highly specialized security datasets, leads to "attention dispersion". Under adversarial conditions, the multi-head attention mechanism fails to aggregate global semantic features effectively when the input sequence is disrupted by non-standard tokens. Convolutional models, in particular 2_Word_CNN (Acc = 0.9621) and 6_Word_ResNet (Acc = 0.9630), showed consistently high results, indicating their ability to effectively identify local n-gram patterns characteristic of direct injections.

Of particular scientific interest is the performance of models at the character level. The 7_Char_CNN model achieved an accuracy of 0.9631, which is practically identical to the best word models, despite the fact that it uses only 23.873 parameters (54 times less than 2_Word_CNN). This proves that the information contained in morphological character sequences is sufficient to identify most types of attacks even without using dictionary embeddings [21,22]. To visually evaluate the resolution of classifiers and compare their ability to operate with a low false positive rate, a combined ROC curve graph was constructed in Figure 2.

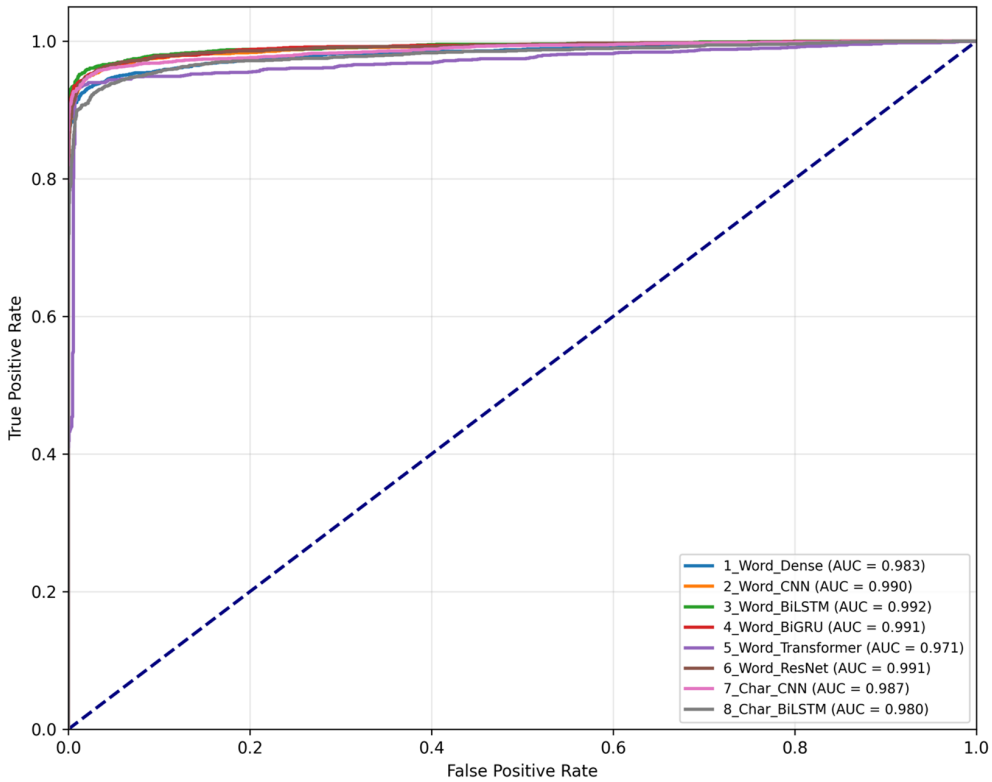

Figure 2. Combined ROC plot for comparing model resolution. The diagonal dashed line represents the performance of a random classifier (baseline, AUC = 0.5).

Visual analysis of the ROC curves in Figure 2 demonstrates that BiLSTM and Char_CNN maintain high sensitivity even at a strict operational constraint of FPR < 0.01. For live systems, this translates to minimal user disruption while maintaining a robust security posture against prompt manipulation. This indicates high sensitivity of the models even at very low false positive rate (FPR < 0.01) values. Mathematically, this means that the probability distributions for legitimate and malicious queries have minimal overlap in semantic space. The AUC value of 0.9918 for BiLSTM indicates near-perfect class separation capability in the absence of adversarial noise. The transformer model curve is significantly lower than other architectures, visualizing the performance gap found in this dataset. The distinction between word-level and character-level models in this graph is minimal for clean data, but the situation changes dramatically when analyzing computational costs and training time.

The next stage of the study was to analyze computational complexity and inference efficiency. Since attack detection must occur in real time, the processing time per query and convergence speed were considered key indicators of practical applicability. A summary of the time and resource costs of the models is provided in Table 4.

Table 4. Computational costs and dynamics of learning architectures.

Model	Accuracy	F1	Time (Tr)	Latency (ms)	Params	Epochs
1_Word_Dense	0.9551	0.9551	29.3	0.0289	1,282,113	24
2_Word_CNN	0.9621	0.9621	17.5	0.0425	1,302,657	5
3_Word_BiLSTM	0.9681	0.9681	133.1	0.3088	1,354,369	5
4_Word_BiGRU	0.9613	0.9613	126.1	0.2506	1,338,241	5
5_Word_Transformer	0.9190	0.9185	136.5	0.3078	1,301,153	6
6_Word_ResNet	0.9630	0.9629	20.9	0.0592	1,308,929	5
7_Char_CNN	0.9631	0.9631	127.6	0.0993	23,873	22
8_Char_BiLSTM	0.9599	0.9599	1581.3	0.7757	21,953	34

Table 4 shows a sharp difference in inference latency and training time, which is not always directly proportional to the number of parameters. The 1_Word_Dense architecture provides the fastest response (0.0289 ms), but its accuracy is inferior to the leaders. The optimal balance of performance and accuracy is demonstrated by 2_Word_CNN (0.0425 ms), making it the preferred choice for ultra-low latency systems. In contrast, recurrent models are significantly slower: 8_Char_BiLSTM requires 0.7757 ms per query, which is 26 times slower than the fastest model. This latency overhead (0.7757 ms for 8_Char_BiLSTM) is a direct consequence of the $O(n)$ sequential complexity of recurrent units operating over an extended character-level horizon ($L_c = 1000$). While this represents a significant increase in latency compared to the 1_Word_Dense model, it constitutes a justifiable "security-performance trade-off" for reinforcing high-value cryptographic interfaces where symbolic robustness is more critical than sub-millisecond throughput.

From a practical deployment perspective, the choice between architectures follows a "cost-risk" logic: while 2_Word_CNN offers the lowest operational cost (0.0425 ms latency), it fails to protect against symbolic obfuscation. In contrast, 8_Char_BiLSTM, despite higher latency, demonstrates enhanced resilience suitable for filtering inputs to high-value assets (such as KMS/PKI workflows), where the risk of a successful injection outweighs the computational overhead.

Particular attention should be paid to the number of training epochs: word-level models (CNN, ResNet, and BiLSTM) converge extremely quickly, in 5–6 epochs. This is because embedding words provides a strong semantic signal at the beginning. In contrast, character models (Char_CNN, Char_BiLSTM) require significantly more iterations (22–34 epochs) because they have to learn the feature hierarchy from raw character sequences [18]. The 8_Char_BiLSTM model exhibited the highest training cost (1581.3 s), reflecting the complexity of learning hierarchical character representations. However, its memory efficiency (21,953 parameters) makes it uniquely suitable for secure edge-inference where privacy must be ensured within constrained hardware environments. This complex trade-off between speed, accuracy, and model size is visualized in the bubble chart of performance in Figure 3.

Figure 3 illustrates the existence of three different performance clusters. The lower left sector is occupied by Dense and CNN models, which provide high speed with stable accuracy, reflected by small bubbles. The central area belongs to recurrent models and transformers. The size of the bubbles clearly demonstrates the advantage of symbolic architectures in terms of memory resource savings, making them ideal for Edge devices. However, from a system design perspective, the optimal 'protection point' on the graph is shifted towards CNN architectures, which provide sub-millisecond latency without significant loss of classification accuracy. The analysis also shows that the 6_Word_ResNet model with a latency of 0.0592 ms is a strong alternative to the baseline CNN, as residual connections allow for higher accuracy with minimal computational slowdown.

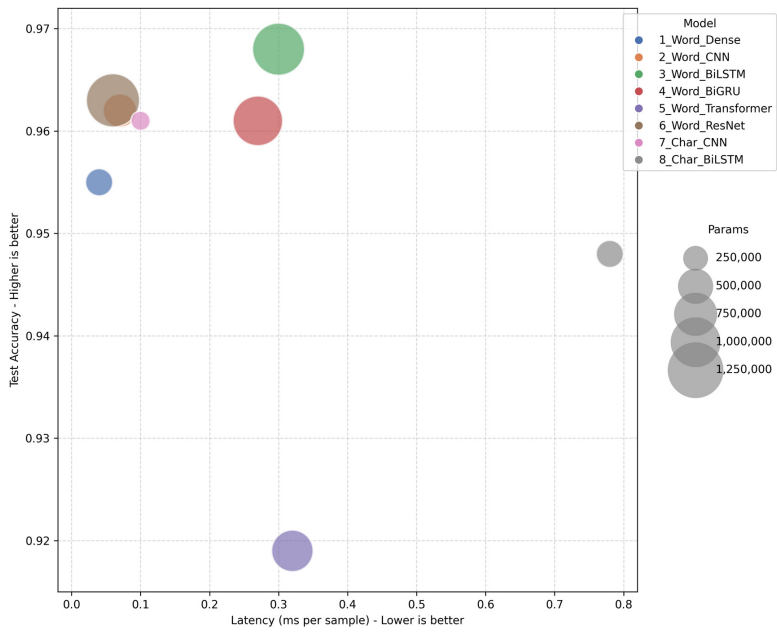

Figure 3. Bubble chart of model performance: accuracy vs. latency vs. parameters.

The most important part of our results is the evaluation of the models' robustness to adversarial attacks based on independent and purposefully noisy samples. We define adversarial robustness as the model's accuracy retention under specific perturbation functions $\epsilon(x)$. Specifically, to address the reviewer's concern regarding the formalization of attack vectors, we define the variants presented in Table 5 as follows:

1. Spacing Attack ($\epsilon_{spacing}$): A perturbation function where a whitespace delimiter is inserted between every character of the input token sequence t, transforming a semantic unit like "select" into "s e l e c t". This targets tokenizer relying on subword merging, forcing the model to process the input as a sequence of independent characters.
2. Homoglyph Attack ($\epsilon_{homoglyph}$): A substitution function mapping Latin characters to their visual Cyrillic or Greek counterparts (e.g., 'a' (U+0061) $\rightarrow$ 'a' (U+0430)). This disrupts the byte-level representation of the prompt while preserving its visual semantics for the human observer, effectively bypassing filter lists based on exact string matching.

Table 5. Report on model resilience to adversarial attacks on independent datasets.

Model	Acc Jailbreaks	Acc Malicious Deep	Acc Forbidden Qs	Acc Malignant	Acc Prediction Guard	Acc Safe Quora	False Positive Rate	Robustness Homoglyph	Robustness Spacing	Robustness Mixed Case
1_Word_Dense	0.9865	0.7871	1	0.2087	0.9282	0.982	0.018	0.61	1	0.99
2_Word_CNN	0.9923	0.9163	1	0.2062	0.9666	0.9742	0.0258	0.22	1	0.98
3_Word_BiLSTM	0.9913	0.943	1	0.2119	0.9676	0.9784	0.0216	0.98	1	0.99
4_Word_BiGRU	0.9923	0.9468	1	0.2347	0.9698	0.9678	0.0322	0.98	1	0.99
5_Word_Transformer	0.9802	0.749	1	0.172	0.8455	0.9961	0.0039	0.97	0.61	0.98
6_Word_ResNet	0.9865	0.9202	1	0.1872	0.9563	0.9886	0.0114	0.02	0.99	0.98
7_Char_CNN	0.9947	0.749	1	0.2732	0.9344	0.9916	0.0084	0.99	0.89	0.99
8_Char_BiLSTM	0.9807	0.6122	1	0.5471	0.9076	0.9904	0.0096	0.98	1	0.98

We simulated these scenarios, along with Unicode character substitution and Mixed Case perturbations, using heterogeneous datasets to test the models' generalization capabilities. A complete report on the reliability of the models is presented in Table 5.

A detailed analysis of Table 5 reveals the fundamental vulnerability of word-level models to structural distortions in the text. While all architectures successfully identify explicit "forbidden" topics (Acc_Forbidden_Qs = 1.0), the introduction of adversarial noise in the Malignant dataset reveals a critical "failure mode" for word-level models. For instance, the 5_Word_Transformer's accuracy collapses to 0.1720, and the 3_Word_BiLSTM to 0.2119. This precipitous decline occurs because word-level tokenizers map OOV tokens generated by character substitutions to generic placeholders, causing the model to lose the semantic chain required for classification. This phenomenon highlights the failure of traditional tokenization to maintain semantic integrity under character-level perturbations, where attention patterns disperse and fail to capture the malicious intent. In contrast, the 8_Char_BiLSTM character model maintains an accuracy of 0.5471, which is 2.5 times higher than that of word models. This quantitative result empirically validates that character representation enables the model to bypass tokenization-based manipulations by operating on invariant morphological features.

The most impressive result is the Robustness_homoglyph metric (resistance to Unicode character replacement). The 2_Word_CNN and 6_Word_ResNet models proved to be almost entirely vulnerable to homoglyph attacks, with robustness scores ρ of 0.22 and 0.02, respectively. This confirms that architectures relying heavily on local n-gram patterns or residual mappings of word embeddings cannot maintain integrity when a single character substitution disrupts the expected token hash. This indicates that deep word architectures become 'blind' when an attacker replaces at least one character in a keyword. At the same time, character models (7_Char_CNN, 8_Char_BiLSTM) maintain a stability of 0.98–0.99, proving their ability to see the internal structure of a character regardless of encoding. In terms of privacy and false positive rates (FPRs), the best results were shown by 5_Word_Transformer (0.0039) and character architectures (0.0084–0.0096), which minimizes the risk of unauthorized blocking of legitimate queries in the Safe_Quora dataset. To analyze the balance between detection security and privacy, a final trade-off plot was constructed, as shown in Figure 4.

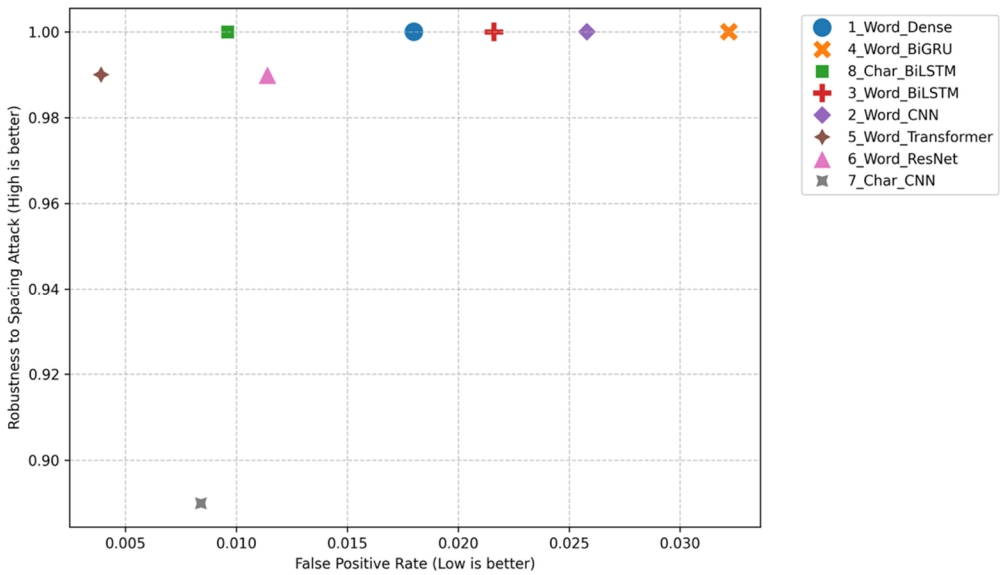

Figure 4. Security–privacy trade-off graph (FPR versus reliability).

The graph in Figure 4 clearly distinguishes between models: symbolic architectures confidently occupy the ideal balance sector (low FPR and high stability). Conversely, word-level models are grouped in the zone of high vulnerability to combined manipulations despite their high semantic accuracy in laboratory tests. Analysis of the complex Malicious_Deep dataset showed that word models (4_Word_BiGRU, 0.9468) better understand the deep semantics of attacks compared to character models (0.6122), indicating the existence of a fundamental trade-off: word-level models better understand 'what' is said (semantics), while character-level models understand 'how' the attack is carried out (morphology).

The distribution of results on the Prediction_Guard set ($Acc \approx 0.90$–0.96) confirms the high generalization ability of trained detectors to detect attempts to leak system prompts. However, the critical decline in accuracy on the Malignant dataset to the level of random guessing for most word models is an alarming signal for LLM security system developers. This proves that to ensure the integrity of intelligent systems, it is necessary to implement hybrid approaches or give preference to character-based architectures.

Summarizing the experimental data obtained, three main scientific theses can be identified:

- The 4_Word_BiGRU architecture is the most accurate for analyzing complex hidden semantics and malicious instructions in normalized form [14,22].
- Character-level vectorization, specifically in the 8_Char_BiLSTM architecture, provides a "robust fallback" layer, maintaining a detection rate of 0.5471 on highly obfuscated payloads—a 2.5× improvement over the best word-level counterparts.
- Hardware acceleration allows these neural network systems to be implemented with a latency of about 1 ms, making them practically applicable for real-time query filtering without compromising the performance of the underlying LLM.

The resulting data provide a solid scientific basis for the development of multi-level, next-generation protection systems capable of adapting to the changing cyber threat landscape and ensuring a stable level of user data privacy in today's intelligent ecosystems.

5. Discussion

Our comparative analysis of eight deep learning architectures reveals a complex landscape of security–performance trade-offs, where the choice of an optimal detector is governed not only by accuracy but by its specific resilience to adversarial symbolic noise and its impact on system-level privacy. A thorough examination of the collected data reveals that the prevailing approach to evaluating LLM security systems, predominantly reliant on "clean" laboratory datasets, is inherently constrained and poses significant risks when applied to genuine cyber threats. From a representation learning perspective, our findings demonstrate that the inductive bias of the evaluated architectures fundamentally dictates their resilience to adversarial noise, identifying word-level tokenization as a bottleneck for robust feature extraction. Even though the models demonstrated high convergence during MPDD corpus training, their behavior under adversarial conditions revealed critical vulnerabilities often overlooked in standard content moderation studies [1,6,7].

The primary conclusion of the study is the validation of the efficacy of recurrent structures, particularly the 3_Word_BiLSTM model, which attained a maximum accuracy of 0.9681 and an Area Under the Curve (AUC) of 0.9918 in standard assessments. This success can be attributed to the efficacy of bi-directional memory in modeling semantic dependencies between initial system instructions and user input. In typical PIA attack scenarios, such as the "Ignore previous instructions and perform an X" action, the critical threat is often obscured in the final part of the sentence, while the prefix may contain a completely legitimate context. By analyzing the sequence in both directions, bidirectional long short-term memory (BiLSTM) captures these relationships more effectively than

convolutional networks, which are constrained by the size of the convolutional core. The subpar performance of the 5_Word_Transformer ($\rho_{spacing} = 0.61$) is theoretically grounded in the attention dispersion phenomenon. From a representation learning perspective, the self-attention mechanism lacks the inherent sequence inductive bias found in RNNs. When confronted with symbolic noise (spacing or homoglyphs), the attention heads fail to aggregate meaningful features from fragmented tokens, leading to a "focus collapse" where the model cannot distinguish malicious intent from background noise [10,15].

The limitation of sequential information loss was most evident in the 1_Word_Dense model. Despite achieving an accuracy of 0.9551 on clean data, its AUC and F1 scores on aggressive adversarial samples indicate shallow learning. The global average pooling operation transforms the input query into a static semantic "bag-of-words" representation, where the temporal order of tokens is irretrievably lost. This makes the model completely unable to distinguish between a situation where the word 'ignore' is used in a legitimate context (e.g., discussing errors in program code) and a situation where it is part of a destructive command to take control of a large language model. This highlights the need to use architectures that preserve the temporal or structural sequence of the query, even if this leads to a slight increase in inference latency. Convolutional neural networks (CNNs) took the middle position, demonstrating F1 scores of 0.9621 (Word) and 0.9631 (Char). Their effectiveness is due to the ability of filters to detect local n-grams characteristic of malicious commands. However, CNNs are less successful at capturing global dependencies, making them vulnerable to attacks where the malicious instruction is blurred by long context or separated by numerous neutral insertions, which is a critical risk when processing repeated requests to complex key management systems (KMSs) [7,10,11].

A critical aspect of the study was the analysis of resistance to adversarial attacks, which revealed a fundamental structural vulnerability in models relying on lexical tokenization. The most striking and alarming result is the drop in the 6_Word_ResNet model to a level of 0.02 resistance to homoglyph attacks and the 2_Word_CNN to 0.22. This means that simply replacing visually identical Latin letters with their Cyrillic counterparts completely blinds these defense systems. This vulnerability is an architectural byproduct of subword tokenization (e.g., BPE or WordPiece): the tokenizer maps words containing homoglyphs to OOV indices or generic $\langle UNK \rangle$ tokens. This effectively 'blinds' word-level models, as the semantic integrity of the input is destroyed before reaching the neural layers. The fact that the transformer loses nearly 40% of its effectiveness in a simple interval attack (resilience score of 0.61) illustrates a fundamental flaw in evaluating security models based solely on normalized laboratory data. The ability to bypass word-level filters via simple character-level obfuscation highlights a potential vulnerability in systems where LLMs serve as natural language interfaces for PKI and KMS workflows. Such adversarial actions could theoretically compromise the integrity of the workflow logic by manipulating the model's context to attempt unauthorized API calls [5,11]. In a bounded threat model, this poses a risk not to the cryptographic primitives themselves, but to the semantic access control layer: a successful injection could plausibly induce an LLM agent to generate parameters for a signing request to a Hardware Security Module (HSM). This scenario suggests a pathway for an attacker to bypass the semantic intent verification logic, effectively coercing the automated system into processing a transaction using valid credentials but invalid intent.

In contrast, character models (7_Char_CNN and 8_Char_BiLSTM) demonstrated exceptional robustness to structural perturbations. Char_BiLSTM maintained a stability score of 0.98 for homoglyphs and 1.00 for Spacing attacks ($\epsilon_{spacing}$) in most scenarios. Character-level architectures effectively treat adversarial noise as additional symbolic features rather than OOV events. This intrinsic robustness allows the model to preserve the morphological signature of malicious intent even under heavy spacing or homoglyph substitution [11,24].

The order of characters remains stable, allowing the network to identify the characteristics of a malicious word through the syntax of its letters. Moreover, character models performed best on the most challenging Acc_Malignant dataset, where Char_BiLSTM achieved an accuracy of 0.5471, which is 2.5 times higher than all word models without exception. Our results suggest character-level vectorization as a critical component of privacy-by-design for intelligent systems, providing a supplementary layer for data minimization and the mitigation of personally identifiable information (PII) exfiltration [2,14]. This superior robustness comes with a significant latency penalty: the 8_Char_BiLSTM model exhibits a latency of 0.7757 ms—an 18-fold increase compared to convolutional models. However, within a layered defense-in-depth framework, this overhead is justifiable for high-value assets (KMS/PKI), where the risk of a successful injection outweighs the microsecond delays in pre-filtering [3,11].

Computational efficiency and scalability are critical factors for practical deployment in systems that process millions of queries every day. In our research, convolutional neural networks (CNNs) demonstrated 5–7 times faster execution speed with only 1–2% loss in pure data accuracy. For real-time systems that require a response in less than 0.1 ms, 2_Word_CNN (0.0425 ms) or 7_Char_CNN (0.0993 ms) are the most rational choices. It is also worth noting the number of training epochs: word models converged in 5–6 epochs, while character models required 22–34 epochs. This is because the network must learn linguistic rules and morphology from scratch, operating solely on the alphabet. This process produces more flexible and robust representations that do not depend on the stability of a particular tokenizer vocabulary and are resilient to sophisticated attempts to bypass semantic access controls [3,11,16].

An important aspect that is often overlooked is the trade-off between security and user privacy, which is expressed through the false positive rate (FPR). The 5_Word_Transformer model showed the lowest FPR (0.0039), making it the most 'tolerant' of clean data. However, this advantage is completely negated by even minor adversarial perturbations. Character models, in particular 8_Char_BiLSTM, provide a consistently low FPR (0.0096) while maintaining high robustness. In real-world deployment, this means that character models will less frequently block requests from innocent users while maintaining high quality of service. We propose a "cost-risk" deployment strategy: high-stakes environments like KMS require aggressive detection thresholds (minimizing Type II errors), whereas general-purpose applications should prioritize low false positive rates (e.g., 0.0039 achieved by Transformer) to maintain user experience. Integrating these detectors into CI/CD pipelines allows for automated policy enforcement and real-time monitoring of trust boundaries [1,2].

Analysis of the ability to generalize external datasets (Jailbreaks, Forbidden Qs, and Safe Quora) confirmed that architectures trained on MPDD successfully detect known attack vectors. However, results on the Malicious_Deep dataset revealed a fundamental theoretical gap. BiGRU and BiLSTM word models significantly outperformed character models on this dataset, indicating their excellent ability to perform deep semantic analysis of complex linguistic constructions. This creates a 'detection paradox': models that best capture high-level semantics ("what" is said) are most vulnerable to structural perturbations ("how" it is encoded). Conversely, character models are adept at recognizing morphological attacks but may overlook subtle semantic manipulations. This finding aligns with theoretical works that demonstrate a higher degree of representation variability, typified by increased regularity and diminished semantic intricacy [2,11,14]. These representations demonstrate reduced reliance on word distribution in the training data, indicating the possibility of enhanced transfer and application in varied contexts.

Consistent with the "trustworthy AI" narrative, we advocate for a layered defense-in-depth framework where character-level BiLSTM serves as a robust pre-filter. This

architecture must be integrated into CI/CD pipelines for continuous model monitoring and enforcement of governance policies to ensure compliant LLM deployment in enterprise environments [12,16,17]. The protection of cryptographic primitives against masked attacks requires systems to change classification thresholds dynamically by using statistical data anomalies for detection.

Future research should focus on hybrid, adaptive defense layers that integrate character-level robustness with RAG-based context sanitization and proactive "red-teaming" feedback loops. While the current models are static, moving toward online threshold tuning and adaptive filtering remains a priority for countering evolving obfuscation tactics like Base64 or ROT13. The primary objective is the development of proactive systems capable of detecting malicious intent regardless of linguistic obfuscation. Ensuring the integrity of interfaces accessing public keys and private user data stands as a high priority for protecting future intelligent ecosystems from adversarial threats [2,7,8,11].

The complete analysis of the obtained results indicates that ensuring the confidentiality and security of artificial intelligence necessitates the adoption of comprehensive reliability indicators, superseding the use of simplified accuracy indicators. Our results, specifically the contrast between ResNet's vulnerability ($\rho = 0.02$) and Char-BiLSTM's resilience (Acc = 0.5471 on Malignant), underscore that character-level modeling is not merely an alternative but a prerequisite for securing intelligent systems against symbolic evasion. The implemented architecture serves as a dependable defense layer designed to mitigate harmful attacks. Specifically, the system aims to filter unauthorized access attempts to critical functions and sensitive information. The study findings reveal that LLMs require two essential elements to function within enterprise key management systems and PKI, which include cryptographic reliability and deep neural network reliability for input instruction processing. Symbolic models used as pre-filters serve as a fundamental security measure. This measure reinforces the security posture of systems interacting with HSMs by detecting unauthorized command patterns within compromised LLM interfaces, serving as a critical validation layer against semantic manipulation.

It is important to note that this study primarily evaluates detection accuracy and adversarial robustness of the classifiers. While robust detection serves as a critical control within a broader defense-in-depth strategy, it does not mathematically guarantee the elimination of cryptographic risks. The reduction in systemic risk in PKI/KMS environments relies on the integration of these detectors with other controls, such as strict output policies and hardware-level enforcement, which are beyond the scope of this classification benchmark.

6. Conclusions

This study provides a systematic evaluation of eight heterogeneous neural architectures for the preemptive detection of PIAs within LLM security pipelines. Our findings indicate that industry reliance on word-level transformer architectures is insufficient for robust adversarial detection. Despite their semantic depth, these models lack the inductive bias needed to withstand low-level text perturbations, rendering them unreliable as standalone defenses. Consequently, the choice of defensive architecture must go beyond laboratory accuracy, prioritizing a cost-risk assessment that balances inference latency and adversarial resilience against the requirements of critical infrastructures, such as KMS or PKI.

A salient finding of this study is that transformer models, despite their widespread adoption and state-of-the-art status in general natural language processing tasks, exhibit specific limitations when integrated into adversarial attack detection systems. The 5_Word_Transformer model exhibited a significant performance collapse under spacing perturbations ($\rho = 0.61$), confirming that attention-based designs lacking explicit sequence

inductive bias fail to maintain representation stability. From a representation learning perspective, this "attention dispersion" prevents the model from identifying malicious intent when the structural integrity of tokens is compromised. This indicates that, under typical circumstances, more than 39% of malicious instructions that have been successfully identified bypass security filters simply by inserting spaces between letters. In contrast, the 8_Char_BiLSTM architecture demonstrates a significantly more stable security profile, exhibiting an accuracy of 0.9599 and maintaining robustness scores of 1.0 for spacing attacks and 0.98 for homoglyph substitutions, effectively neutralizing the most common structural obfuscation vectors. This outcome underscores the pivotal benefit of symbol-based analysis in contrast to token-based analysis in the context of countering adaptive adversaries.

The most significant finding of the study was the recognition of the fundamental importance of representations at the symbol level as a foundation for constructing reliable protective contours. The efficacy of character-level models is attributable to the characteristics of input signal processing. Adversarial attacks, such as Spacing and Homoglyph, take advantage of the vulnerabilities of word-level tokenizers, which disassemble familiar words into sequences of unknown $\langle UNK \rangle$ tokens, thereby completely dismantling the semantic network query vector. In the case of 6_Word_ResNet, resistance to homoglyph attacks was critically low ($\rho = 0.02$). This collapse is a direct consequence of word-level tokenization, where visually similar but technically distinct Unicode characters force the tokenizer to generate $\langle UNK \rangle$ (unknown) tokens, effectively "blinding" the residual layers to the malicious payload. At the character level, the addition of spaces or replacement of Unicode characters does not result in the introduction of elements that are entirely unfamiliar to the system. Instead, the characters undergo a simple shuffling or replacement by graphically similar counterparts that the recurrent BiLSTM model can learn to ignore through the analysis of the internal morphological structure of the word. This property renders character-level approaches significantly more resilient to character transformations, while preserving detection integrity under conditions of active entanglement.

Furthermore, it was determined that character models demonstrated a substantially superior capacity for generalization when applied to independent datasets. The 8_Char_BiLSTM model demonstrated a minimal decline in accuracy, ranging from 1% to 3%, when evaluated on external corpora such as Jailbreaks, Forbidden Questions, and Safe Quora. In contrast, word-level architectures, including Transformer, exhibited a more substantial decline, ranging from 8% to 15%, on complex adversarial datasets, as evidenced in Malignant. This indicates that word-level models are prone to overfitting to a specific attack vocabulary represented in the training set, whereas character-based architectures learn more general, abstract patterns of malicious text construction, allowing security system developers to extrapolate lab test results to real-world data streams in product environments with greater confidence.

For practical deployment in high-security intelligent systems, the character-level BiLSTM architecture is proposed as a viable solution that balances detection accuracy ($F1 \approx 0.96$) with exceptional adversarial robustness (0.98–1.0) and acceptable computational cost. Although the inference latency of this model is around 0.77 ms (higher than that of dense models), it remains orders of magnitude below the typical token generation latency of modern LLMs (often >20 ms per token), rendering it effectively imperceptible to the end user. For systems with extreme latency requirements (less than 0.1 ms), we recommend using Char_CNN, which retains over 96% of the leader's accuracy at significantly lower computational cost. At the same time, we highlight the comparative inefficiency of transformer models for this specific high-load task: their need for GPU resources and vulnerability to symbolic noise render them less optimal compared to convolutional neural networks, which have 60 times fewer parameters (approximately 22k versus 1.3M).

We advocate for a layered defense-in-depth architecture: (1) a normalization layer; (2) a robust character-level filter (Char_BiLSTM) to neutralize symbolic noise; and (3) a high-level semantic validator (4_Word_BiGRU) for complex intent analysis. This hierarchy aims to ensure that protection is maintained even when one layer is bypassed by sophisticated obfuscation. While not a standalone solution for cryptographic security, this structure significantly strengthens the defense posture of natural language interfaces to critical operations (e.g., KMS workflows), reducing the surface area for unauthorized command generation via LLM intermediaries. The total latency of such a pipeline will not exceed 5 ms, which is ideal for integration into API gateways.

An important aspect discussed in detail in the article is the impact of false positive results (false positive rate) on privacy. A system that is overly aggressive in blocking (which is typical for BiLSTM and BiGRU based on words with an FPR of up to 0.032) paradoxically reduces security by driving users to less secure platforms. Our results show that character-based models provide higher reliability and better privacy (FPR < 0.01). In order to provide a more nuanced approach to this issue, we recommend the implementation of context-dependent thresholds. Specifically, we suggest the use of context-aware thresholds: lenient thresholds (e.g., $p > 0.3$) for open systems to prioritize usability, and stricter criteria for critical infrastructure and PKI, where minimizing False negatives is paramount.

For organizations with limited resources, 7_Char_CNN (latency 0.099 ms, homoglyph protection 0.99) is the only viable option. For organizations that possess sufficient infrastructure, the 8_Char_BiLSTM model should be prioritized. The predominant conclusion for the industry is that security is not a static metric of accuracy, but rather a dynamic process of adaptation. The ResNet model's critical vulnerability to homoglyphs, with a recorded robustness score of 0.02, suggests a need for reassessing existing security protocols that rely solely on embedding-based detection.

At the industry level, it is advisable to transition from isolated initiatives to coordinated standards in a minimum of three dimensions. Initially, the necessity arises to open dynamic datasets for LLM attacks, which are subject to regular updates in order to circumvent the "outdated" threat model characteristic of static datasets such as HackAPrompt. Secondly, there is a necessity to formalize the metrics of reliability and latency for defensive components. Such metrics should include attack success rate and integrated adversary resilience assessments. Thirdly, it is imperative that mandatory adversarial testing (red teaming) prior to LLM system deployment become the prevailing standard, superseding its current status as an optional 'pentest' for marketing reports.

Despite the significant accuracy achieved in this research, its limitations must be considered. First, our evaluation focuses on the classification layer of the security pipeline; while high detection accuracy correlates with reduced risk, it does not mathematically guarantee the prevention of all downstream exploits in a fully integrated PKI/KMS system. Second, the study focuses on Latin and Cyrillic scripts; the efficacy of character-level modeling for logographic languages (e.g., Chinese) remains an open question requiring distinct validation. Finally, latency measurements were tied to a specific Apple Silicon hardware profile, which may vary across cloud (CUDA) or legacy (CPU-only) environments.

Future research should explore obfuscation methods beyond character-level perturbations, including Base64, ROT13, and cryptographic-inspired techniques, and develop hybrid detection frameworks that integrate real-time feedback, adaptive policies, and privacy-preserving mechanisms to enhance protection against evolving adversarial strategies.

The trade-offs involved in architectural choices concerning the detection of prompt injections are of critical importance. Higher accuracy of clean data is often accompanied by lower resistance to attacks; greater complexity is associated with higher costs; and over-detection poses a threat to privacy. It is important to note that no architecture is universally applicable;

however, the symbolic approaches presented in this work currently provide the most balanced foundation for building reliable intelligent ecosystems. Without a transition toward robust, symbol-aware detection integrated into CI/CD pipelines, the global AI landscape remains vulnerable to evolving manipulations that threaten data privacy and system integrity. The joint efforts of science and industry are essential to creating a future where LLMs are more resilient against malicious interference targeting users' confidential data.

Author Contributions: Conceptualization, O.K. and S.Y.; methodology, O.K. and R.S.; software, O.K.; validation, O.K., S.Y. and M.K.; formal analysis, O.K. and M.K.; investigation, O.K. and R.S.; resources, O.K. and S.Y.; data curation, O.K.; writing—original draft preparation, O.K.; writing—review and editing, S.Y., R.S. and M.K.; visualization, O.K.; supervision, S.Y.; project administration, S.Y.; funding acquisition, R.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are openly available in the Kaggle repository at https://www.kaggle.com/datasets/mohammedaminejebbar/malicious-prompt-detection-dataset-mpdd/ (accessed on 2 October 2025).

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Mirtaheri, S.L.; Movahed, N.; Shahbazian, R.; Pascucci, V.; Pugliese, A. Cybersecurity in the Age of Generative AI: A Systematic Taxonomy of AI-Powered Vulnerability Assessment and Risk Management. *Future Gener. Comput. Syst.* **2026**, *175*, 108107. [CrossRef]
2. Yan, B.; Li, K.; Xu, M.; Dong, Y.; Zhang, Y.; Ren, Z.; Cheng, X. On Protecting the Data Privacy of Large Language Models (LLMs) and LLM Agents: A Literature Review. *High-Confid. Comput.* **2025**, *5*, 100300. [CrossRef]
3. Trabilsy, M.; Prabha, S.; Gomez-Cabello, C.A.; Haider, S.A.; Genovese, A.; Borna, S.; Wood, N.; Gopala, N.; Tao, C.; Forte, A.J. The PIEE Cycle: A Structured Framework for Red Teaming Large Language Models in Clinical Decision-Making. *Bioengineering* **2025**, *12*, 706. [CrossRef] [PubMed]
4. Kang, D.; Li, X.; Stoica, I.; Guestrin, C.; Zaharia, M.; Hashimoto, T. Exploiting Programmatic Behavior of LLMs: Dual-Use Through Standard Security Attacks. In Proceedings of the 2024 IEEE Security and Privacy Workshops (SPW), San Francisco, CA, USA, 23 May 2024; pp. 132–143. [CrossRef]
5. Balakrishnan, P.; Leema, A.A. Vulnerabilities and Defenses: A Monograph on Comprehensive Analysis of Security Attacks on Large Language Models. *Indian J. Inf. Sources Serv.* **2025**, *15*, 442–467. [CrossRef]
6. Vassilev, A.; Oprea, A.; Fordyce, A.; Anderson, H. *Adversarial Machine Learning: A Taxonomy and Terminology of Attacks and Mitigations*; National Institute of Standards and Technology (U.S.): Gaithersburg, MD, USA, 2024; p. NIST 100-2e2023. [CrossRef]
7. Greshake, K.; Abdelnabi, S.; Mishra, S.; Endres, C.; Holz, T.; Fritz, M. Not What You've Signed Up for: Compromising Real-World LLM-Integrated Applications with Indirect Prompt Injection. In Proceedings of the 16th ACM Workshop on Artificial Intelligence and Security, Copenhagen, Denmark, 30 November 2023; Association for Computing Machinery: New York, NY, USA, 2023; pp. 79–90. [CrossRef]
8. Miceli Barone, A.V.; Sun, Z. A Test Suite of Prompt Injection Attacks for LLM-Based Machine Translation. In Proceedings of the Ninth Conference on Machine Translation, Miami, FL, USA, 15–16 November 2024; Association for Computational Linguistics: Miami, FL, USA, 2024; pp. 380–450. [CrossRef]
9. Kantor, I. Practical Attacks on LLMs: Full Guide. Available online: https://iterasec.com/blog/practical-attacks-on-llms/ (accessed on 28 December 2025).
10. Hung, K.-H.; Ko, C.-Y.; Rawat, A.; Chung, I.-H.; Hsu, W.H.; Chen, P.-Y. Attention Tracker: Detecting Prompt Injection Attacks in LLMs. In *Findings of the Association for Computational Linguistics: NAACL 2025*; Chiruzzo, L., Ritter, A., Wang, L., Eds.; Association for Computational Linguistics: Albuquerque, NM, USA, 2025; pp. 2309–2322. [CrossRef]
11. Lan, Q.; Kaul, A.; Jones, S. Prompt Injection Detection in LLM Integrated Applications. *Int. J. Netw. Dyn. Intell.* **2025**, *4*, 100013. [CrossRef]
12. Shvetsova, O.; Katalshov, D.; Lee, S.-K. Innovative Guardrails for Generative AI: Designing an Intelligent Filter for Safe and Responsible LLM Deployment. *Appl. Sci.* **2025**, *15*, 7298. [CrossRef]

13. Sayeedi, M.F.A.; Bin Hossain, M.; Hassan, M.K.; Afrin, S.; Hossain, M.M.S.; Hossain, M.S. JailbreakTracer: Explainable Detection of Jailbreaking Prompts in LLMs Using Synthetic Data Generation. *IEEE Access* **2025**, *13*, 123708–123723. [CrossRef]
14. Omri, S.; Abdelkader, M.; Hamdi, M. SafetyRAG: Towards Safe Large Language Model-Based Application Through Retrieval-Augmented Generation. *J. Adv. Inf. Technol.* **2025**, *16*, 243–250. [CrossRef]
15. Shi, L.; Kang, Y.; Hu, J.; Li, X.; Yang, M. Meticulous Thought Defender: Fine-Grained Chain-of-Thought (CoT) for Detecting Prompt Injection Attacks of Large Language Models. *IEEE Access* **2025**, *13*, 113194–113207. [CrossRef]
16. Rai, P.; Sood, S.; Madisetti, V.K.; Bahga, A. GUARDIAN: A Multi-Tiered Defense Architecture for Thwarting Prompt Injection Attacks on LLMs. *J. Softw. Eng. Appl.* **2024**, *17*, 43–68. [CrossRef]
17. Hadiprakoso, R.B.; Wilujengning, W.; Amiruddin, A. Adaptive Multi-Layer Framework for Detecting and Mitigating Prompt Injection Attacks in Large Language Models. *J. Inf. Syst. Eng. Bus. Intell.* **2025**, *11*, 473–487. [CrossRef]
18. LLM01:2025 Prompt Injection. Available online: https://genai.owasp.org/llmrisk/llm01-prompt-injection/ (accessed on 28 December 2025).
19. Rao, V.S.; Kumar, A.; Lakkaraju, H.; Shah, N.B. Detecting LLM-Generated Peer Reviews. *PLoS ONE* **2025**, *20*, e0331871. [CrossRef]
20. Milani, A.; Franzoni, V.; Florindi, E.; Omarbekova, A.; Bekmanova, G.; Yergesh, B. When AI Is Fooled: Hidden Risks in LLM-Assisted Grading. *Educ. Sci.* **2025**, *15*, 1419. [CrossRef]
21. Jain, B.; Pawar, P.; Gada, D.; Patwa, T.; Kanani, P.; Patil, D.; Kurup, L. Comprehensive Analysis of Machine Learning and Deep Learning Models on Prompt Injection Classification Using Natural Language Processing Techniques. *Int. Res. J. Multidiscip. Technovation* **2025**, 24–37. [CrossRef]
22. Ergün, A.; Onan, A. Adversarial Prompt Detection in Large Language Models: A Classification-Driven Approach. *Comput. Mater. Contin.* **2025**, *83*, 4855–4877. [CrossRef]
23. Evaluating Prompt Injection Datasets. Available online: https://hiddenlayer.com/innovation-hub/evaluating-prompt-injection-datasets/ (accessed on 28 December 2025).
24. Malicious Prompt Detection Dataset (MPDD). Available online: https://www.kaggle.com/datasets/mohammedaminejebbar/malicious-prompt-detection-dataset-mpdd/ (accessed on 28 December 2025).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.