

Comparative Analysis of Transformer Models for Prompt Injection Detection in Autonomous Large Language Model Agents

Bhoomi Roy ^{1,✉}, Arbaz ¹, Sujeet Kumar ¹, Shivam Kumar ¹, Rahul Verma ¹

1. Computer Science and Engineering, Greater Noida Institute of Technology, Greater Noida, IND

Received: June 30, 2026 | Review began: September 02, 2026 | Review ended: September 30, 2026 | Published: October 20, 2026

© Copyright 2026

This is an open access article distributed under the terms of the Creative Commons Attribution License CC-BY 4.0., which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Abstract

Prompt injection attacks pose a significant security challenge for large language model systems, particularly autonomous large language model-based agents that may interact with external tools, application programming interfaces, and data sources. This study presents a comparative evaluation of five pre-trained transformer models, namely BERT, RoBERTa, DeBERTa, DistilBERT, and ALBERT, fine-tuned for binary prompt injection classification. A balanced dataset of 4,224 samples was compiled from 10 publicly accessible corpora containing benign and injection prompts and divided into training (2,956), validation (634), and test (634) sets. The models were trained in the Kaggle environment using an NVIDIA Tesla T4 GPU, the AdamW optimizer, and a linear warm-up scheduling strategy. On the held-out test set, RoBERTa achieved the highest classification performance, with 97.48% accuracy, 97.52% macro-precision, 97.48% macro-recall, 97.48% macro-F1-score, and 99.64% area under the receiver operating characteristic curve. DistilBERT achieved the highest reported throughput of 819.13 samples/s and the lowest per-sample latency of 1.22 ms. Statistical analysis using the Wilcoxon signed-rank test and SHapley Additive exPlanations-based interpretability analysis were also conducted to examine model differences and influential linguistic features. The findings demonstrate that fine-tuned transformer classifiers can achieve strong performance on the evaluated benchmark dataset while exhibiting different accuracy-efficiency characteristics. However, the results should be interpreted within the scope of the evaluated data distribution, as the study does not include independent external datasets or adaptive adversarial attacks. These limitations indicate that further validation is required to establish generalizability and robustness in real-world prompt-injection defense scenarios.

Categories: The Adversarial AI, Machine Learning (ML), Security and Privacy

Keywords: prompt injection detection, large language models, transformer models, machine learning, security in ai agents

Introduction

There has been an almost unprecedented proliferation of large language models (LLMs) in commercial, research, and enterprise applications, providing high-level natural language understanding and generation capabilities and enabling autonomous reasoning in natural language. At the same time, a class of adversarial inputs called prompt injection attacks has emerged as a security concern. A prompt injection attack is an attack that tricks an AI model into executing an unauthorized action by placing instructions in the prompt sequence that cause the AI to break or violate its policies [1]. Such attacks occur mostly in agentic LLM applications, where models are given access to external application programming interfaces, databases, and tools that are not effectively supervised by human agents [2].

How to cite this article:

Roy B, . A, Kumar S, et al. (October 20, 2026) Comparative Analysis of Transformer Models for Prompt Injection Detection in Autonomous Large Language Model Agents. Cureus J Comput Sci 3 : es44389-026-00329-y. DOI <https://doi.org/10.7759/s44389-026-00329-y>

Currently, there are two general classes of defense strategies: input filtering and model-level defenses. Initial prompt injection sanitization techniques were not robust against semantically different or obfuscated injections. Later work focused on machine learning-based classifiers applied to prompt representations. Using traditional models based on term frequency features and linear classifiers provided an intuitive yet limited baseline. With the introduction of pre-trained transformer models, the ability to identify adversarial patterns with much greater fidelity and nuance improved through the use of contextually rich representations.

Although there has been a rise in studies on the topic, most available studies focus on a single transformer model or only a couple of models. Therefore, systematic research into the trade-offs among accuracy, computational efficiency, and model size across different transformer architectures remains limited. To address these gaps, this work comprehensively compares five widely used transformer architectures (BERT, RoBERTa, DeBERTa, DistilBERT, and ALBERT). The five models were fine-tuned using a representative and balanced multi-source prompt injection dataset. The measures of model performance include classification accuracy, macro-averaged precision, recall, F1-score, area under the receiver operating characteristic curve (ROC-AUC), and false negative rate (FNR). Computational efficiency metrics, such as inference time and throughput, are further analyzed to support deployment decision-making. The Wilcoxon signed-rank test was used to determine the statistical significance of performance differences, and SHapley Additive exPlanations (SHAP) analysis was performed to assess model interpretability.

Related work

Prompt injection defenses in the early years of the threat mainly relied on input sanitization and rule-based filtering techniques to process attacks. Sayeedi et al. [1] offered a remedy by proposing JailbreakTracer, an explainable detection framework that leverages synthetic data generation to raise the alarm on jailbreaking prompts targeting LLMs. To complement this detection-centric research, Derner et al. [2] proposed a taxonomy of security risks associated with prompt-based interactions with LLMs, offering a systematic categorization of the threat landscape.

A broad survey of the wider field of adversarial machine learning in text is also relevant here. Alsmadi et al. [3] conducted a thorough literature review focusing on adversarial attacks and defenses in natural language processing (NLP) pipelines, while Malik et al. [4] conducted a systematic review of adversarial attack types, defensive controls, and technologies used in machine learning systems in general. Similar methodological parallels can be observed in adjacent security domains: Yan et al. [5] surveyed adversarial attack and defense approaches for malware classification, and Seneviratne et al. [6] used self-supervised vision transformers for malware detection tasks, demonstrating the success of transformer-based representations in adversarial detection tasks beyond pure NLP. Akhtar et al. [7] provided a broad overview of attacks and defenses in computer vision, laying the groundwork that has since informed the study of adversarial attacks and defenses in the text domain. Sadrizadeh et al. [8] further illustrated a block-sparse adversarial attack that exploits transformer-based text classifiers, showing that even fine-tuned transformer text classifiers retain exploitable vulnerabilities.

Comparative studies of structural classification tasks, such as prompt injection detection, have been conducted extensively. Adoma et al. [9] tested BERT, RoBERTa, DistilBERT, and XLNet on a text-based emotion recognition task, finding significant performance differences between architectures on the same task. This comparative approach has been extended to security-related applications: Otieno et al. [10] classified phishing emails using the BERT transformer model, and Rifat et al. [11] assessed phishing text using BERT for social engineering attack detection. Karim et al. [12] presented a hybrid machine learning-based phishing detection system utilizing URL features that combined learned and classical representations, while Patra et al. [13] carried out a comparative analysis of different pre-trained BERT variants for phishing URL detection. Outside the security context but methodologically comparable, Danish et al. [14] conducted sentiment analysis research comparing the performance of several models on cryptocurrency-related textual data using BERT, RoBERTa, and ChatGPT, while Hazim and Ata [15] studied BERT and RoBERTa alongside logistic regression and neural network baselines to determine the authenticity of textual data in the age of artificial intelligence.

How to cite this article:

Roy B, . A, Kumar S, et al. (October 20, 2026) Comparative Analysis of Transformer Models for Prompt Injection Detection in Autonomous Large Language Model Agents. *Cureus J Comput Sci* 3 : es44389-026-00329-y. DOI <https://doi.org/10.7759/s44389-026-00329-y>

Materials And Methods

Dataset construction

Ten publicly accessible corpora were collected from Hugging Face and represented heterogeneous categories of benign and prompt-injection text. Injection-oriented samples were obtained from Lakera/gandalf_ignore_instructions, rubend18/ChatGPT-Jailbreak-Prompts, deepset/prompt-injections, and xTRam1/safe-guard-prompt-injection, while benign samples were obtained from HuggingFaceH4/no_robots, tatsu-lab/alpaca, databricks/databricks-dolly-15k, OpenAssistant/oasst2, and jackhhao/jailbreak-classification-benign. A balanced sampling procedure was applied to obtain equal numbers of benign and injection samples. The final dataset contained 4,224 samples, comprising 2,112 benign and 2,112 injection prompts. The data sources and sample counts are shown in Table 1.

Source Dataset	Benign Used	Injection Used	Total Used	Type
HuggingFaceH4/no_robots	846	0	846	Benign
Lakera/gandalf_ignore_instructions	0	777	777	Injection
OpenAssistant/oasst2	253	0	253	Benign
databricks/databricks-dolly-15k	168	0	168	Benign
deepset/prompt-injections	32	203	235	Mixed
jackhhao/jailbreak-classification	0	457	457	Injection
jackhhao/jailbreak-classification-benign	55	0	55	Benign
rubend18/ChatGPT-Jailbreak-Prompts	0	76	76	Injection
tatsu-lab/alpaca	254	0	254	Benign
xTRam1/safe-guard-prompt-injection	504	599	1,103	Mixed
Total	2,112	2,112	4,224	Balanced

TABLE 1: Dataset source summary and sample allocation

Open-source code and data hosted on Kaggle [16].

The complete source allocation is presented in Table 2. The dataset was partitioned into training, validation, and test sets using a stratified random split of 70%, 15%, and 15%, respectively, resulting in 2,956 training samples, 634 validation samples, and 634 test samples. Class balance was maintained across all three partitions.

Because the dataset was assembled from multiple publicly available sources, the resulting collection provides source diversity but should not be interpreted as a set of fully independent corpora. The present study does not include a dedicated semantic or near-duplicate analysis across the 10 source datasets; therefore, potential overlap or similarity between samples from different sources cannot be completely excluded. This consideration is important when interpreting the generalizability of the reported test-set performance.

Text-length statistics were also examined. Benign prompts had a mean length of 44.47 words (standard deviation = 75.68), whereas injection prompts had a higher mean length of 82.32 words (standard deviation = 143.01). The dataset and source code are publicly available through the associated Kaggle repository to support reproducibility.

Split	Benign	Injection	Total	Proportion
Training	1,478	1,478	2,956	70%
Validation	317	317	634	15%
Test	317	317	634	15%
Total	2,112	2,112	4,224	100%

TABLE 2: Dataset partition summary

Open-source code and data hosted on Kaggle [16].

Data preprocessing

Text samples shorter than 10 characters and those containing only whitespace were excluded from the dataset.

$$y_i = \begin{cases} 1 & \text{if } x_i \in \mathcal{V}_{\text{inject}} \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

where x_i is the input sample and $\mathcal{V}_{\text{inject}}$ is the set of injection indicator keywords.

Ambiguously labeled source datasets were also used for label normalization based on injection-related indicators such as "inject", "jailbreak", "attack", "malicious", "unsafe", and "harmful". To fit within the GPUs' memory limitations, each prompt was limited to a maximum length of 3,000 characters before tokenization.

$$\tilde{x}_i = x_i[0 : L_{\text{max}}], \quad \text{where } L_{\text{max}} = 3,000 \text{ characters} \quad (2)$$

where $\tilde{x}_i$ is the truncated prompt, $L_{\text{max}} = 3,000$ is the character limit, and $T_{\text{max}} \in \{128, 256\}$ is the model-specific token length.

Model-specific tokenizers were used with padding, according to the maximum model length of 128 or 256 tokens, depending on the model. All preprocessing, data splitting, and model initialization steps were made reproducible by using the fixed seed "42".

How to cite this article:

Roy B, . A, Kumar S, et al. (October 20, 2026) Comparative Analysis of Transformer Models for Prompt Injection Detection in Autonomous Large Language Model Agents. Cureus J Comput Sci 3 : es44389-026-00329-y. DOI <https://doi.org/10.7759/s44389-026-00329-y>

Model architectures

A variety of pre-training strategies, parameter scales, and transformer architectures were chosen to represent five transformer encoder architectures:

BERT-base-uncased is a bidirectional encoder that was trained using masked language modeling and next-sentence prediction on the BookCorpus and English Wikipedia corpora. It has 12 transformer layers, 768 hidden dimensions, 12 attention heads, and 109.48 million parameters. Classification is performed using the CLS token embedding, which is fed through a linear head.

RoBERTa-base is a tuned-up version of BERT that discarded the next-sentence prediction objective and added the following characteristics: dynamic masking, the use of a much larger number of sentences and longer texts for training, and the treatment of sentences as shorter due to byte-pair encoding tokenization. It has the same architecture as BERT, with 124.65 million parameters.

DeBERTa-v3-base uses disentangled attention, which incorporates content and position embeddings, allowing them to interact within the decomposition before being aggregated. Along with the enhanced discriminative capacity provided by ELECTRA-style replaced token detection pre-training in v3, cascading temporal slot tokens further enhances discriminative capacity. The model consists of 184.42 million parameters, making it the largest of the five models tested.

DistilBERT-base-uncased is the distilled version of BERT, trained using knowledge distillation. It has 6 transformer layers and 66.96 million parameters, around 97% of which are shared with common BERT, while retaining most of BERT's performance on the common benchmark tasks. It is compact which allows for much faster inference.

The factorized embedding parameterization, combined with cross-layer parameter sharing, leads to an extremely low number of parameters in ALBERT-base-v2: 11.69 million, equivalent to just 44.59 MB on disk, while maintaining state-of-the-art representation quality and sentence-order prediction pre-training.

Experimental setup

The AdamW optimizer was used with a linear warm-up schedule covering 10% of the total training steps, followed by linear decay. A binary cross-entropy loss function was used for the binary classification task.

$$\eta_t = \begin{cases} \eta_{\max} \cdot \frac{t}{T_{\text{warm}}} & \text{if } t \leq T_{\text{warm}} \\ \eta_{\max} \cdot \frac{T_{\text{total}} - t}{T_{\text{total}} - T_{\text{warm}}} & \text{if } t > T_{\text{warm}} \end{cases} \quad (3)$$

where $\eta_{\max}$ is the peak learning rate, t is the current step, T_{warm} is the warm-up cutoff, and T_{total} is the total training steps.

The cross-entropy loss function was computed for logits representing binary classification.

$$\mathcal{L}_{CE} = -\frac{1}{N} \sum_{i=1}^N [y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)] \quad (4)$$

where $\mathcal{L}_{CE}$ is the cross-entropy loss, y_i is the true label, $\hat{y}_i$ is the predicted probability, and N is the batch size.

For all architectures except DeBERTa, which was trained in full precision because of gradient instability during automatic mixed-precision training, FP16 automatic mixed precision was employed. The best-performing checkpoint for each architecture was selected according to the validation F1-score and subsequently evaluated on the held-out test set.

All experiments were conducted on Kaggle using an NVIDIA Tesla T4 GPU with 14.56 GB of VRAM, CUDA 12.8, PyTorch version 2.10.0+cu128, and 29.55 GB of available CPU RAM. The total training time for the five models was 98.91 minutes. Model-specific hyperparameters are presented in Table 3.

How to cite this article:

Roy B, . A, Kumar S, et al. (October 20, 2026) Comparative Analysis of Transformer Models for Prompt Injection Detection in Autonomous Large Language Model Agents. Cureus J Comput Sci 3 : es44389-026-00329-y. DOI <https://doi.org/10.7759/s44389-026-00329-y>

All preprocessing, dataset splitting, and model initialization procedures were performed using a fixed random seed of 42 to ensure the reproducibility of the reported experimental run. Consequently, the reported results represent a single controlled experimental run rather than an estimate of variability across multiple independently seeded repetitions. The study therefore does not report confidence intervals or cross-seed variance estimates for the model performance metrics. Additional repeated experiments with multiple random seeds would provide stronger evidence regarding the stability of the observed performance differences.

Model	HuggingFace ID	Parameters	Max Len	Learning Rate	Batch	Epochs
BERT	bert-base-uncased	109.48M	256	2e-5	16	4
RoBERTa	roberta-base	124.65M	256	2e-5	16	4
DeBERTa	deberta-v3-base	184.42M	256	1e-5	16	3
DistilBERT	distilbert-base-uncased	66.96M	128	3e-5	32	5
ALBERT	albert-base-v2	11.69M	128	2e-5	32	5

TABLE 3: Model-specific hyperparameter configuration

Open-source code and data hosted on Kaggle [16].

Evaluation metrics

Performance was evaluated using accuracy, macro-averaged precision, macro-averaged recall, macro-averaged F1-score, and ROC-AUC on the test set, which was not used for training.

$$F1_{\text{macro}} = \frac{1}{C} \sum_{c=1}^C \frac{2 \cdot P_c \cdot R_c}{P_c + R_c} \quad (5)$$

where $F1_{\text{macro}}$ is the average F1-score over all classes, P_c is the precision, R_c is the recall for class c , and C is the total number of classes.

Apart from this, the FNR was calculated to determine the percentage of false negatives, given the deployment risks and the high security risks posed by false negatives.

$$FNR = \frac{FN}{FN + TP} \quad (6)$$

where FN represents missed injections and TP represents correctly detected injections.

The three metrics characterizing inference efficiency included per-sample latency (ms) and throughput (samples/s), measured under a fixed inference procedure using batch evaluation. The p-value from the Wilcoxon signed-rank test (alpha = 0.05), applied to the distributions of macro-averaged F1-scores for pairs of models, was used to assess the

How to cite this article:

Roy B, . A, Kumar S, et al. (October 20, 2026) Comparative Analysis of Transformer Models for Prompt Injection Detection in Autonomous Large Language Model Agents. Cureus J Comput Sci 3 : es44389-026-00329-y. DOI <https://doi.org/10.7759/s44389-026-00329-y>

statistical significance of pairwise performance differences. SHAP was used as the partition explainer on 100 representative test samples from RoBERTa to identify token-level attribution patterns that could help distinguish injected from benign inputs.

Results And Discussion

Test set classification performance

The following table shows the final performance of all five models on the test set (Table 4), which is also displayed graphically in Figure 1 and Figure 2. RoBERTa achieved the highest performance across accuracy (97.48%), macro-averaged precision (97.52%), recall (97.48%), F1-score (97.48%), and ROC-AUC (99.64%). The FNR of RoBERTa was 4.10%, meaning that approximately 4 injection attacks were not detected for every 100 attacks. BERT achieved a high accuracy of 96.69% and a ROC-AUC of 99.64%, placing it second overall. The closeness of their ROC-AUC scores across various classification thresholds indicates a close match in discriminative power between the two models. DistilBERT achieved the highest throughput among all models (819.13 samples/s), alongside a competitive accuracy of 96.53%, though its FNR (5.36%) was tied with ALBERT for the highest among the five models. Despite the large gap in model parameters (184.42M vs. 11.69M) and the significant disparity in architectural complexity, both DeBERTa and ALBERT achieved 95.74% accuracy on the test set, indicating that model size is not a linear indicator of classification accuracy beyond a certain threshold. The ROC-AUC values of DeBERTa (98.82%) and ALBERT (99.11%) were still high, showing good probabilistic discrimination despite lower hard-decision accuracy.

Model	Accuracy	Precision	Recall	F1 (Macro)	ROC-AUC	FNR	Latency (ms)	Throughput
RoBERTa	0.9748	0.9752	0.9748	0.9748	0.9964	0.0410	3.563	280.66
BERT	0.9669	0.9674	0.9669	0.9669	0.9964	0.0505	3.908	255.87
DistilBERT	0.9653	0.9660	0.9653	0.9653	0.9938	0.0536	1.221	819.13
DeBERTa	0.9574	0.9574	0.9574	0.9574	0.9882	0.0442	7.632	131.03
ALBERT	0.9574	0.9576	0.9574	0.9574	0.9911	0.0536	3.388	295.12

TABLE 4: Final test set classification performance across all transformer models

Open-source code and data hosted on Kaggle [16].
FNR, False Negative Rate; ROC-AUC, Area Under the Receiver Operating Characteristic Curve

With the highest macro-F1-score of 97.48%, the RoBERTa model outperformed the other four models evaluated, while DistilBERT performed similarly well, with a macro-F1-score of 96.53%. Architectures of widely varying complexity, such as DeBERTa and ALBERT, achieved macro-F1-scores of 95.74%, suggesting that they can define similar classification surfaces for this task. The results are competitive, as shown by the inter-model range in F1-scores of 1.74 percentage points across the full range of model architectures evaluated. This performance ranking follows known results on general NLP benchmarks, where the improved pre-training process provides a measurable improvement for RoBERTa over the original

BERT in downstream text classification tasks. Knowledge distillation showed how DistilBERT could achieve near-BERT accuracy using just 40% of the number of parameters, further underscoring the real-world implications of this classification pipeline for security purposes.

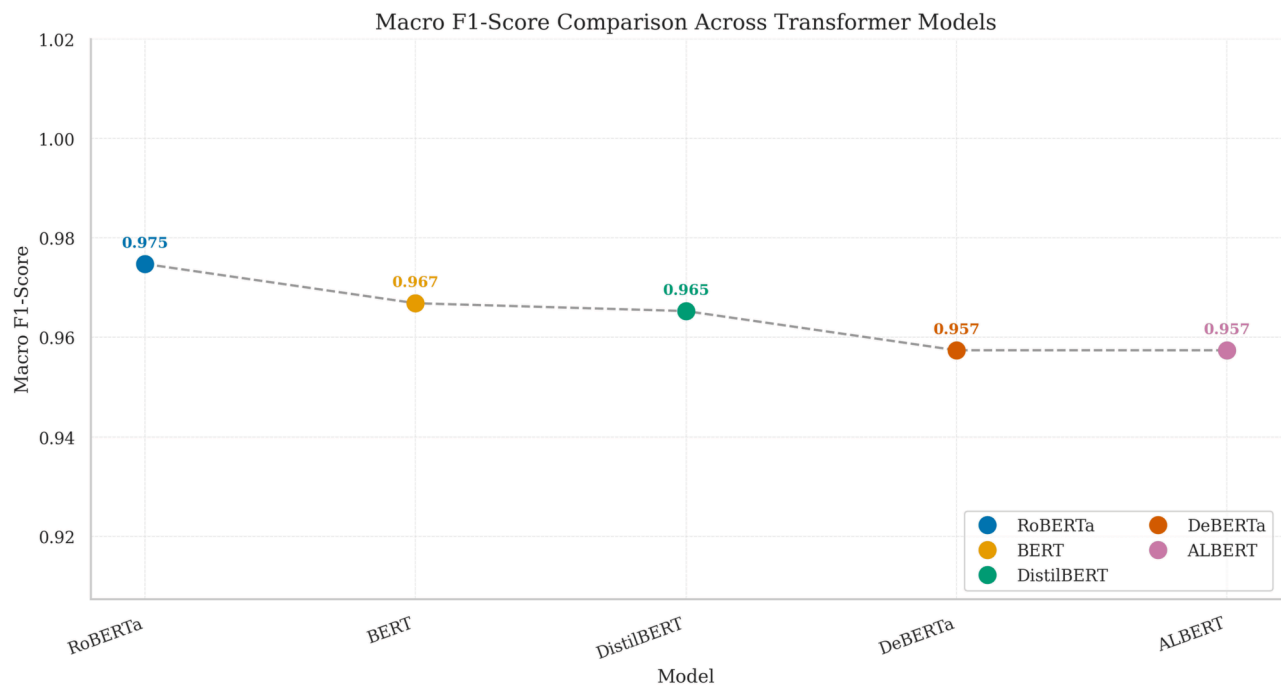


FIGURE 1: Macro-F1-score comparison bar chart across all five models on the test set

Open-source code and data hosted on Kaggle [16].

The discriminative ability of all five transformer models was high, with ROC-AUC scores above 0.988 on the test set, evaluated across all balanced samples. Near-perfect class separability was observed for RoBERTa, which achieved the highest ROC-AUC of 99.64%, equal to that of BERT. DistilBERT achieved an AUC of 99.38%, followed by ALBERT at 99.11% and DeBERTa at 98.82%. The close positioning of all five ROC curves toward the upper-left corner of the ROC space further indicates that the fine-tuned transformer models exhibit strong discriminative capability in distinguishing injection attacks from benign prompts across a wide range of decision thresholds. Although RoBERTa and BERT achieved identical ROC-AUC scores of 99.64%, RoBERTa obtained a higher hard-decision accuracy of 97.48% compared with BERT's 96.69%, representing a 0.79-percentage-point improvement at the selected classification threshold.

How to cite this article:

Roy B, . A, Kumar S, et al. (October 20, 2026) Comparative Analysis of Transformer Models for Prompt Injection Detection in Autonomous Large Language Model Agents. Cureus J Comput Sci 3 : es44389-026-00329-y. DOI <https://doi.org/10.7759/s44389-026-00329-y>

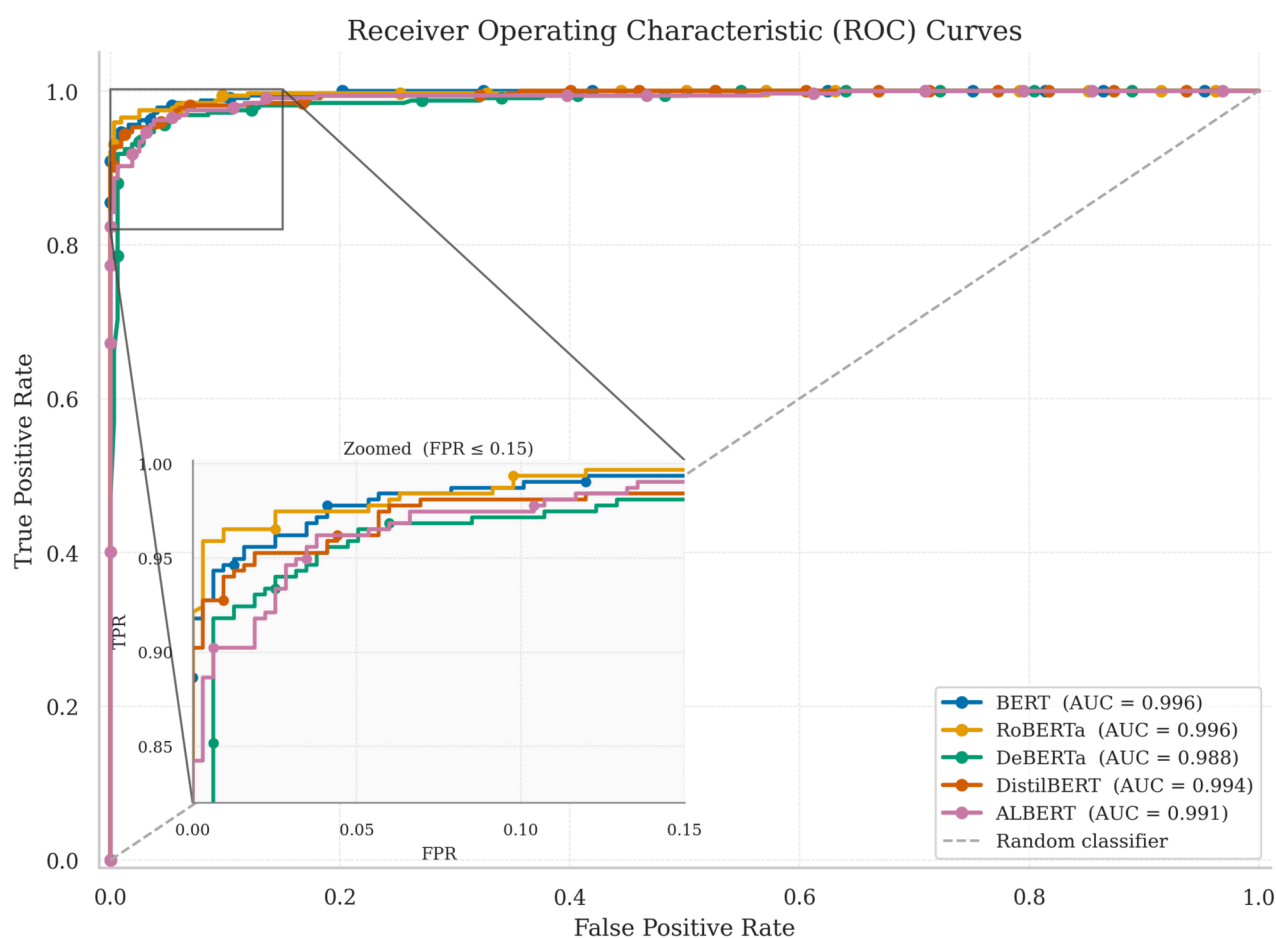


FIGURE 2: ROC curves for all five transformer models on the held-out test set.

Open-source code and data hosted on Kaggle [16].

ROC-AUC, Area Under the Receiver Operating Characteristic Curve

RoBERTa has the least "unbalanced" error distribution across the 634-sample test set, with an estimated rate of misclassifying an injection attack as benign of 4.10% (13 false negatives) and a rate of incorrectly raising a flag on a benign prompt of 3.10% (10 false positives). The false negative count for BERT was slightly higher, at about 16 (Figure 3), indicating slightly lower injection recall compared to DistilBERT, which had a similar profile with about 17 false negatives. In the task of security-critical classification, DeBERTa and ALBERT yielded FNRs of 4.42% and 5.36%, respectively, showing a slight performance advantage for RoBERTa in this relatively fine-grained classification task.

How to cite this article:

Roy B, . A, Kumar S, et al. (October 20, 2026) Comparative Analysis of Transformer Models for Prompt Injection Detection in Autonomous Large Language Model Agents. Cureus J Comput Sci 3 : es44389-026-00329-y. DOI <https://doi.org/10.7759/s44389-026-00329-y>

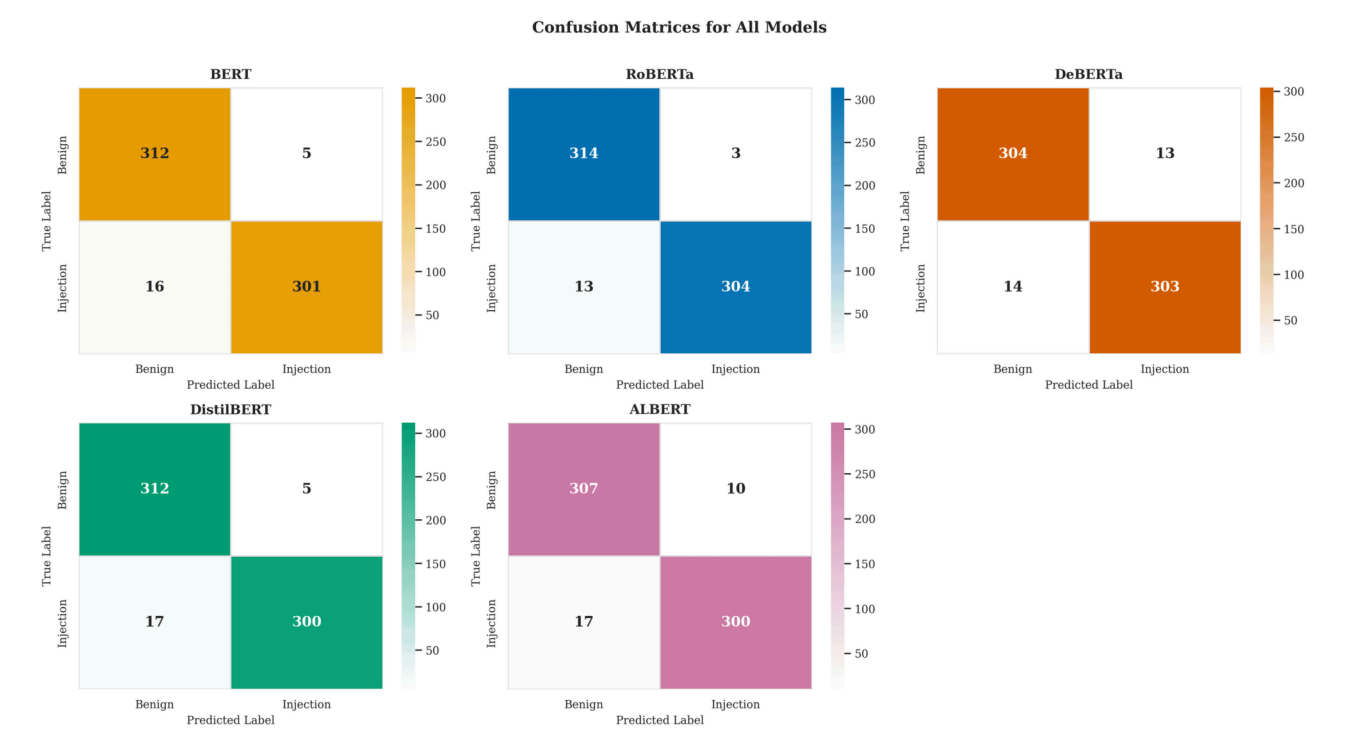


FIGURE 3: Confusion matrices for all five models on the test set, showing true positive, false positive, true negative, and false negative counts

Open-source code and data hosted on Kaggle [16].

RoBERTa had the most even precision-recall balance of all five models, achieving 97.52% precision and 97.48% recall at the macro level, resulting in a macro-F1-score of 97.48%. BERT achieved a precision of 96.74% and a recall of 96.69%, showing a slight bias toward reducing false positives relative to capturing true positives. DistilBERT achieved high precision (96.60%) and recall (96.53%), indicating a good balance in identifying the classification boundary, with no clear directional bias. DeBERTa achieved equal macro-averaged precision and recall of 95.74%, with the benign and injection categories showing a symmetric error distribution. ALBERT showed a marginally higher precision (95.76%) than recall, indicating a slightly conservative classification tendency (Figure 4). The grouped bar visualization again indicates that all five architectures could be suitable for incorporation into a prompt injection detection pipeline, as no single model shows a significant or practically meaningful difference in precision and recall.

How to cite this article:

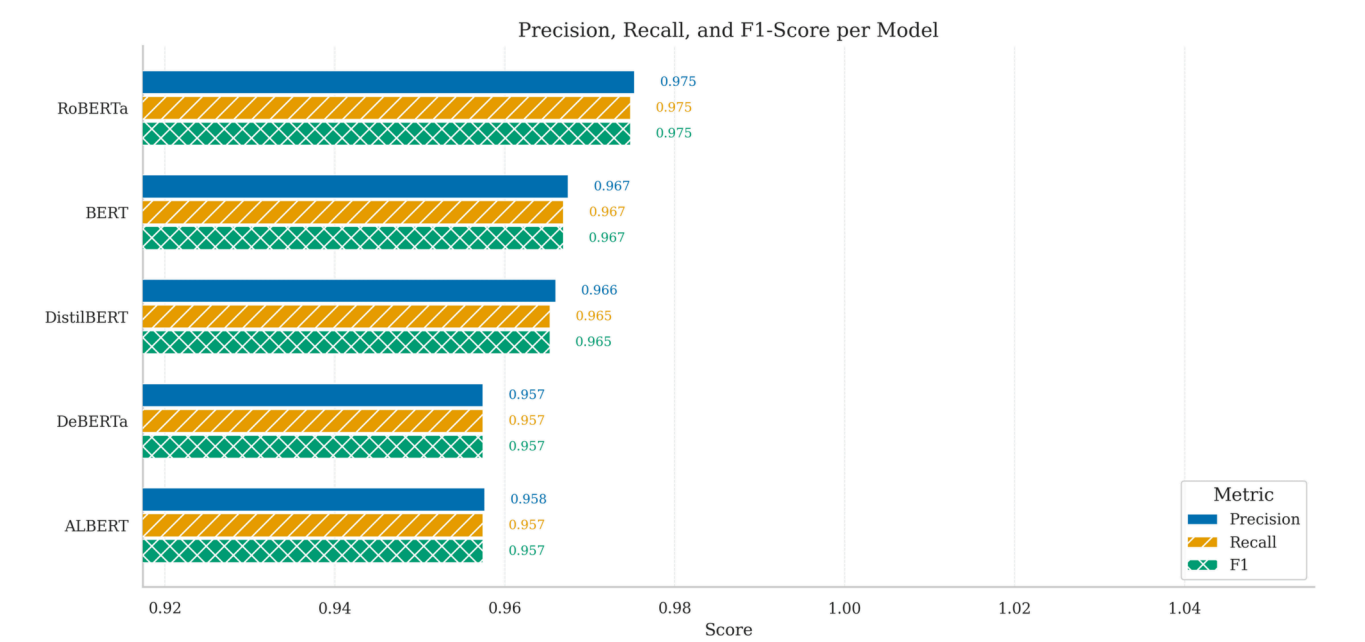


FIGURE 4: Precision, recall, and F1-score-grouped bar chart for all five models

Open-source code and data hosted on Kaggle [16].

Efficiency analysis

The inference efficiency results show a strong inverse relationship between predictive power and the computational resources required per prediction. For deployment scenarios with high query volumes, DistilBERT had the highest throughput at 819.13 samples/s, with a short per-sample latency of 1.22 ms. ALBERT was the most resource-friendly model, with a model size of only 44.59 MB, a throughput of 295.12 samples/s, and a latency of 3.39 ms. The latencies of RoBERTa (3.56 ms) and BERT (3.91 ms) were found to be similar. The disentangled attention mechanism and larger parameter set of DeBERTa led to the lowest throughput (131.03 samples/s) and highest latency (7.63 ms) among all models. The efficiency profiles are displayed in Figure 5 and Figure 6.

How to cite this article:

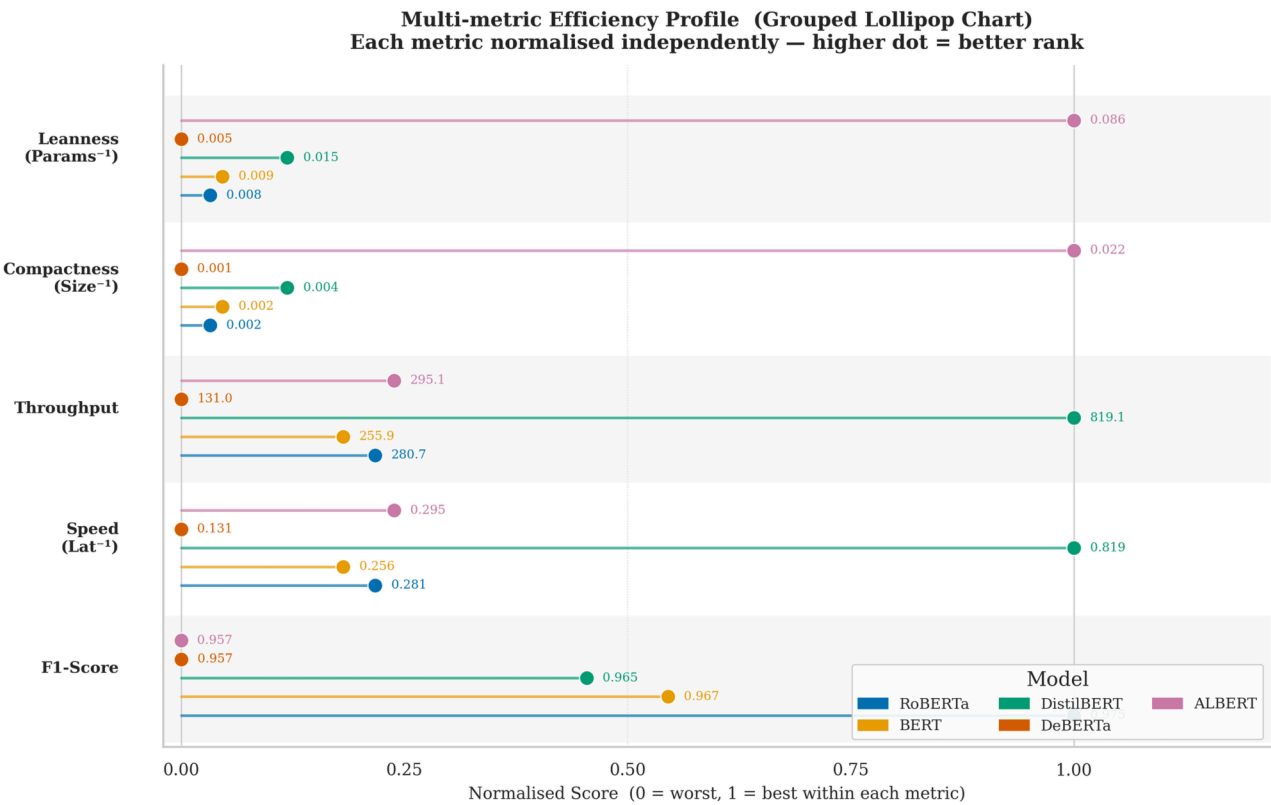


FIGURE 5: Efficiency radar chart comparing normalized throughput, latency, model size, and parameter count across all five models

Open-source code and data hosted on Kaggle [16].

How to cite this article:

Roy B, . A, Kumar S, et al. (October 20, 2026) Comparative Analysis of Transformer Models for Prompt Injection Detection in Autonomous Large Language Model Agents. Cureus J Comput Sci 3 : es44389-026-00329-y. DOI <https://doi.org/10.7759/s44389-026-00329-y>

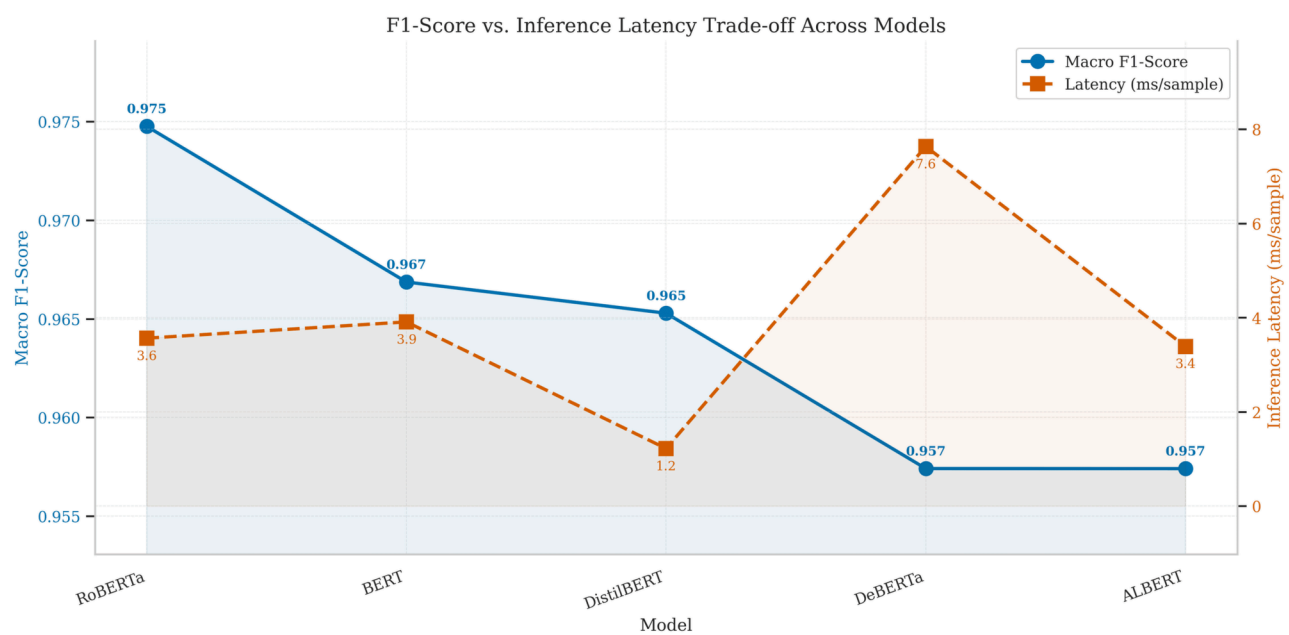


FIGURE 6: Scatter plot of per-sample inference latency versus macro-F1-score, illustrating the accuracy-efficiency trade-off
Open-source code and data hosted on Kaggle [16].

Statistical significance testing

For each inference batch, macro-F1-score distributions were obtained for all five models on the held-out test set, and the 10 possible model pairs were compared using the pairwise Wilcoxon signed-rank test. Table 5 presents the resulting p-value matrix, and Figure 7 provides the corresponding visualization. Among the evaluated model pairs, the comparison between RoBERTa and DeBERTa produced a statistically significant difference $p = 0.0207$, whereas the remaining pairwise comparisons did not reach the significance threshold of $p < 0.05$.

These statistical results should be interpreted within the scope of the current experimental design. The Wilcoxon analysis evaluates differences between the models across batches of the single held-out test set and does not quantify variability across independently repeated training runs. Because the experiments used a fixed random seed of 42, the analysis cannot establish whether the observed performance differences would remain consistent across alternative initialization and data-sampling seeds. Repeated experiments across multiple random seeds, together with confidence intervals or other measures of between-run variability, would provide stronger evidence concerning the stability of the reported model differences.

How to cite this article:

Model	BERT	RoBERTa	DeBERTa	DistilBERT	ALBERT
BERT	1.0000	0.2024	0.5691	0.7508	0.9107
RoBERTa	0.2024	1.0000	0.0207*	0.8811	0.6945
DeBERTa	0.5691	0.0207*	1.0000	0.1613	0.3896
DistilBERT	0.7508	0.8811	0.1613	1.0000	0.5615
ALBERT	0.9107	0.6945	0.3896	0.5615	1.0000

TABLE 5: Pairwise Wilcoxon signed-rank test p-values

Open-source code and data hosted on Kaggle [16].
Asterisk (*) denotes statistical significance at $p < 0.05$.

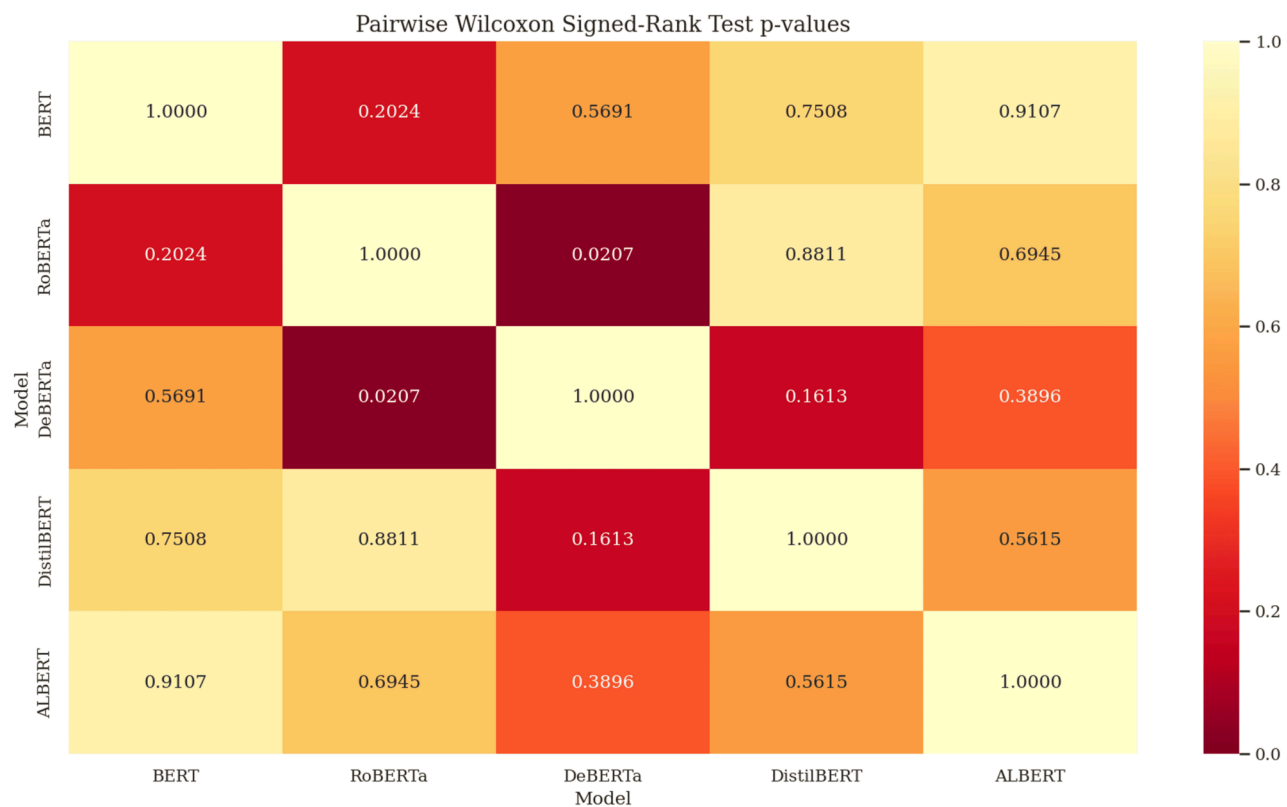


FIGURE 7: Pairwise Wilcoxon signed-rank test p-value heatmap across all five models

Open-source code and data hosted on Kaggle [16].

How to cite this article:

Roy B, . A, Kumar S, et al. (October 20, 2026) Comparative Analysis of Transformer Models for Prompt Injection Detection in Autonomous Large Language Model Agents. Cureus J Comput Sci 3 : es44389-026-00329-y. DOI <https://doi.org/10.7759/s44389-026-00329-y>

Interpretability analysis

The RoBERTa model was used to illustrate token-level attribution patterns through SHAP partition explainer analysis on 100 test samples. The results showed that imperative override language (e.g., "ignore," "forget," "disregard," "instead," and "now") had the strongest positive SHAP scores for the injection category. In contrast, tokens related to user requests, questions, and domain-specific vocabulary were assigned lower or negative attribution values, aligning with patterns generally considered benign in user prompting. The SHAP data are summarized in Figure 8, alongside representative waterfall visualizations.

How to cite this article:

Roy B, . A, Kumar S, et al. (October 20, 2026) Comparative Analysis of Transformer Models for Prompt Injection Detection in Autonomous Large Language Model Agents. Cureus J Comput Sci 3 : es44389-026-00329-y. DOI <https://doi.org/10.7759/s44389-026-00329-y>



FIGURE 8: SHAP summary plot displaying token attribution values for injection detection in RoBERTa, with individual sample waterfall plots for representative injection and benign examples

Open-source code and data hosted on Kaggle [16].

SHAP, SHapley Additive exPlanations

Discussion

The findings indicate that transformer-based classifiers can achieve strong binary prompt-injection classification performance on the evaluated multi-source dataset. All five investigated architectures achieved accuracy above 95.7% on the held-out test set, while RoBERTa obtained the highest accuracy, macro-F1-score, and ROC-AUC among the evaluated models. DistilBERT demonstrated a different performance characteristic by combining competitive classification results with the highest measured throughput and the lowest reported inference latency. These findings illustrate the accuracy-efficiency trade-offs that can arise among transformer architectures when applied to prompt-injection classification.

The results should nevertheless be interpreted in relation to the scope of the evaluation. The test set contains 634 samples and is derived from the same overall collection used to construct the training and validation sets. Although the source collection includes 10 publicly available datasets, the present study does not include an independent external test corpus. In addition, the dataset construction procedure does not report a dedicated semantic near-duplicate analysis across the source datasets. Consequently, the reported test-set performance provides evidence of performance on the evaluated benchmark distribution but does not, by itself, establish generalization to unrelated prompt-injection corpora.

The study also evaluates publicly available prompt-injection patterns without specifically testing adaptive adversarial attacks designed to evade the learned classifiers. This is an important limitation for security applications because an attacker aware of the detection mechanism may modify wording, structure, or semantic expression while preserving the intended malicious instruction. The strong performance observed on the evaluated dataset, therefore, should not be interpreted as evidence that the models provide complete or robust protection against previously unseen or adaptive attacks.

The computational measurements further show meaningful differences in inference efficiency. DistilBERT achieved the highest throughput of 819.13 samples/s and the lowest measured latency of 1.22 ms/sample, while ALBERT provided the smallest model footprint at 44.59 MB. RoBERTa combined strong classification performance with moderate inference costs, whereas DeBERTa produced the highest latency and lowest throughput among the evaluated models. These measurements are useful for characterizing the computational behavior of the models under the specific hardware and inference procedure used in this study; however, they should not be interpreted as universal deployment characteristics because latency and throughput can vary across hardware, batch sizes, sequence lengths, software configurations, and serving environments.

The SHAP analysis identified imperative override expressions such as "ignore," "forget," "disregard," "instead," and "now" among the influential features for the RoBERTa classifier. These findings provide interpretability evidence concerning the linguistic patterns used by the model on the evaluated samples. However, the presence of such features does not imply that these lexical patterns will remain reliable indicators under paraphrased, obfuscated, or previously unseen attacks. Adaptive adversaries may deliberately modify these surface patterns while preserving the underlying intent of an injection.

The study has several limitations that should be considered when interpreting its conclusions. First, the dataset contains 4,224 samples and may not cover the full diversity of prompt-injection attacks encountered in practical LLM-agent systems. Second, the held-out test set originates from the same overall dataset distribution rather than an independent external corpus. Third, a dedicated semantic near-duplicate analysis across the 10 source datasets was not included. Fourth, adaptive adversarial attacks and newly emerging attack patterns were not evaluated. Fifth, the experiments used a fixed random seed of 42 and, therefore, the reported results do not quantify cross-seed variability or provide confidence intervals. Finally, the statistical analysis is based on batch-level performance distributions from a single experimental run.

How to cite this article:

Roy B, . A, Kumar S, et al. (October 20, 2026) Comparative Analysis of Transformer Models for Prompt Injection Detection in Autonomous Large Language Model Agents. *Cureus J Comput Sci* 3 : es44389-026-00329-y. DOI <https://doi.org/10.7759/s44389-026-00329-y>

Future studies should address these limitations through independent external evaluation, systematic duplicate and near-duplicate analysis, repeated experiments across multiple random seeds, confidence-interval reporting, adaptive adversarial testing, and evaluation of newly emerging prompt-injection patterns.

Overall, the results provide an empirical comparison of five transformer architectures for binary prompt-injection classification under the experimental conditions studied. The findings support the use of transformer-based classifiers as a promising research direction for detection, while additional validation is required before their performance can be interpreted as evidence of robust or production-ready security protection for autonomous LLM agents.

Conclusions

This study presented a comparative evaluation of five pre-trained transformer models - BERT, RoBERTa, DeBERTa, DistilBERT, and ALBERT - for binary prompt-injection classification in autonomous LLM-agent scenarios. Using a balanced dataset of 4,224 samples collected from 10 publicly available sources, the study evaluated classification performance using accuracy, macro-averaged precision, macro-averaged recall, macro-averaged F1-score, ROC-AUC, and false negative rate, together with inference latency and throughput. On the held-out 634-sample test set, all five models achieved accuracy above 95.7%. RoBERTa obtained the highest accuracy of 97.48%, macro-averaged F1-score of 97.48%, and ROC-AUC of 99.64%, while DistilBERT achieved the highest measured throughput of 819.13 samples/s and the lowest per-sample latency of 1.22 ms. ALBERT provided the smallest model footprint at 44.59 MB. The Wilcoxon signed-rank analysis identified a statistically significant difference between RoBERTa and DeBERTa at $p = 0.0207$, while the remaining evaluated pairwise comparisons did not reach statistical significance at $p < 0.05$. SHAP analysis further identified imperative and instruction-override language as influential features for the RoBERTa classifier.

These findings demonstrate that fine-tuned transformer models can achieve strong binary prompt-injection classification performance under the dataset and experimental conditions investigated. However, the results should not be interpreted as establishing robustness against real-world adaptive adversaries or previously unseen attack patterns. The evaluation was performed on a fixed test set drawn from the same overall data distribution, did not include an independent external dataset or a dedicated semantic near-duplicate analysis across source datasets, and used a fixed random seed rather than repeated, independently seeded experiments. Accordingly, the reported results provide benchmark-based evidence of model performance rather than definitive evidence of generalization or production-level security effectiveness. Future work should evaluate the models on independent external datasets, incorporate adaptive and adversarial prompt-injection attacks, examine newly emerging attack patterns, and investigate multi-label detection of distinct injection categories. Repeated experiments using multiple random seeds, together with confidence intervals and additional uncertainty analysis, would also strengthen the reliability of model comparisons. Ensemble and continuously updated detection approaches may further improve robustness against changing attack distributions. Such extensions would help determine the extent to which transformer-based prompt-injection classifiers generalize beyond the benchmark conditions examined in this study.

Additional Information

Author Contributions

All authors have reviewed the final version to be published and agreed to be accountable for all aspects of the work.

Concept and design: Bhoomi Roy, Arbaz ., Shivam Kumar

Acquisition, analysis, or interpretation of data: Bhoomi Roy, Arbaz ., Sujeet Kumar, Rahul Verma, Shivam Kumar

Drafting of the manuscript: Bhoomi Roy, Arbaz ., Shivam Kumar

Critical review of the manuscript for important intellectual content: Bhoomi Roy, Arbaz ., Sujeet Kumar, Rahul Verma, Shivam Kumar

How to cite this article:

Roy B, . A, Kumar S, et al. (October 20, 2026) Comparative Analysis of Transformer Models for Prompt Injection Detection in Autonomous Large Language Model Agents. Cureus J Comput Sci 3 : es44389-026-00329-y. DOI <https://doi.org/10.7759/s44389-026-00329-y>

Supervision: Sujeet Kumar

Disclosures

Human subjects: All authors have confirmed that this study did not involve human participants or tissue.

Animal subjects: All authors have confirmed that this study did not involve animal subjects or tissue.

Conflicts of interest: In compliance with [the ICMJE uniform disclosure form](#), all authors declare the following:

- **Payment/services info:** All authors have declared that no financial support was received from any organization for the submitted work.
- **Financial relationships:** All authors have declared that they have no financial relationships at present or within the previous three years with any organizations that might have an interest in the submitted work.
- **Other relationships:** All authors have declared that there are no other relationships or activities that could appear to have influenced the submitted work.

Data Availability Statements

The data (and/or code) supporting this study are openly available in Kaggle at <https://www.kaggle.com/code/arbazdevhive/transformer-models-for-prompt-injection>.

Acknowledgements

This article was previously presented at the International Conference on Progressive Trends in Engineering Science and Management (PTESM 2026), held on 10th–11th April 2026. The dataset, source code, trained models, and study materials are publicly available through the project's public Kaggle repository to support transparency and reproducibility. Bhoomi Roy and Arbaz contributed equally to this work and should be considered co-first authors. The authors sincerely acknowledge the valuable guidance, encouragement, and continuous support of Mr. Sujeet Kumar Sir, whose guidance was instrumental in the successful completion of this research.

References

1. Sayeedi MFA, Hossain MB, Hassan MK, Afrin S, Hossain MMS, Hossain MS: [JailbreakTracer: Explainable detection of jailbreaking prompts in LLMs using synthetic data generation](#). IEEE Access. 2025, 13:123708-123723. [10.1109/access.2025.3579996](#)
2. Derner E, Batistič K, Zahálka J, Babuška R: [A security risk taxonomy for prompt-based interaction with large language models](#). IEEE Access. 2024, 12:126176-126187. [10.1109/access.2024.3450388](#)
3. Alsmadi I, Aljaafari N, Nazzal M, et al.: [Adversarial machine learning in text processing: A literature survey](#). IEEE Access. 2022, 10:17043-17077. [10.1109/access.2022.3146405](#)
4. Malik J, Muthalagu R, Pawar PM: [A systematic review of adversarial machine learning attacks, defensive controls, and technologies](#). IEEE Access. 2024, 12:99382-99421. [10.1109/access.2024.3423323](#)
5. Yan S, Ren J, Wang W, Sun L, Zhang W, Yu Q: [A survey of adversarial attack and defense methods for malware classification in cyber security](#). IEEE Communications Surveys & Tutorials. 2022, 25:467-496. [10.1109/comst.2022.3225137](#)
6. Seneviratne S, Shariffdeen R, Rasnayaka S, Kasthuriarachchi N: [Self-supervised vision transformers for malware detection](#). IEEE Access. 2022, 10:103121-103135. [10.1109/access.2022.3206445](#)
7. Akhtar N, Mian A, Kardan N, Shah M: [Advances in adversarial attacks and defenses in computer vision: A survey](#). IEEE Access. 2021, 9:155161-155196. [10.1109/access.2021.3127960](#)
8. Sadrizadeh S, Dolamic L, Frossard P: [Block-sparse adversarial attack to fool transformer-based text classifiers](#). ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Singapore. 2022, 7837-7841. [10.1109/ICASSP43922.2022.9747475](#)

How to cite this article:

Roy B, . A, Kumar S, et al. (October 20, 2026) Comparative Analysis of Transformer Models for Prompt Injection Detection in Autonomous Large Language Model Agents. Cureus J Comput Sci 3 : es44389-026-00329-y. DOI <https://doi.org/10.7759/s44389-026-00329-y>

9. Adoma AF, Henry NM, Chen W: [Comparative analyses of BERT, RoBERTa, DistilBERT, and XLNet for text-based emotion recognition](#). 2020 17th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP), Chengdu, China. 2021, 117-121. [10.1109/ICCWAMTIP51612.2020.9317379](#)
10. Otieno DO, Namin AS, Jones KS: [The application of the BERT transformer model for phishing email classification](#). 2023 IEEE 47th Annual Computers, Software, and Applications Conference (COMPSAC), Torino, Italy. 2023, 1303-1310. [10.1109/COMPSAC57700.2023.00198](#)
11. Rifat N, Ahsan M, Chowdhury M, Gomes R: [BERT against social engineering attack: Phishing text detection](#). 2022 IEEE International Conference on Electro Information Technology (eIT), Mankato, MN, USA. 2022, 1-6. [10.1109/eIT53891.2022.9813922](#)
12. Karim A, Shahroz M, Mustofa K, Belhaouari SB, Joga SRK: [Phishing detection system through hybrid machine learning based on URL](#). IEEE Access. 2023, 11:36805-36822. [10.1109/access.2023.3252366](#)
13. Patra C, Giri D, Maitra T, Kundu B: [A comparative study on detecting phishing URLs leveraging pre-trained BERT variants](#). 2024 6th International Conference on Computational Intelligence and Networks (CINE), Bhubaneswar, India. 2024, 1-6. [10.1109/CINE63708.2024.10881521](#)
14. Danish M, Amjad M, Ahmad T, Salau AO: [Comparative sentiment analysis of cryptocurrency apps using BERT, RoBERTa, and ChatGPT](#). 2024 IEEE 5th International Conference on Electro-Computing Technologies for Humanity (NIGERCON), Ado Ekiti, Nigeria. 2024, 1-5. [10.1109/NIGERCON62786.2024.10927251](#)
15. Hazim LR, Ata O: [Textual authenticity in the AI era: Evaluating BERT and RoBERTa with logistic regression and neural networks for text classification](#). 2024 International Symposium on Electronics and Telecommunications (ISETC), Timisoara, Romania. 2024, 1-6. [10.1109/ISETC63109.2024.10797291](#)
16. [Transformer Models for Prompt Injection](#). (2026). Accessed: June 29, 2026: <https://www.kaggle.com/code/arbazdevhive/transformer-models-for-prompt-injection>.

How to cite this article:

Roy B, . A, Kumar S, et al. (October 20, 2026) Comparative Analysis of Transformer Models for Prompt Injection Detection in Autonomous Large Language Model Agents. Cureus J Comput Sci 3 : es44389-026-00329-y. DOI <https://doi.org/10.7759/s44389-026-00329-y>