iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 **Issue:** VI **Month of publication:** June 2025

DOI: https://doi.org/10.22214/ijraset.2025.72523

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue VI June 2025- Available at www.ijraset.com

A Survey on Emoji and Unicode-Based Data Masking Attacks on AI Systems

Aditi Sakpal
Computer Science Engineering Department, Sikkim Manipal Institute of Technology

Abstract: ***With the constant introduction of AI technologies into our lives, attack surfaces are emerging on the basis of vulnerabilities in AI processing systems. Text input manipulation, through the use of Unicode-based data obfuscation (e.g. the use of emojis, zero-width characters, and homoglyph replacements) is one of them. such techniques enable attackers to bypass filters, evade content moderation, and even insert adversarial prompts into large language models (LLMs). This survey studies the situation regarding emoji and Unicode-based data masking attacks, categorizes the techniques used, determines legitimate uses, and discusses the difficulties with defending against these attacks. Future work is also suggested in the research as well as summarizing possible defence schemes. This paper has tried to assemble recent research and field-based case studies in an effort to form a concrete basis of research.***

Keywords: Unicode, emojis, data masking, adversarial prompts, large language models (LLMs), cybersecurity, prompt injection.

I. INTRODUCTION

Artificial Intelligence (AI) machines have become common in various sectors, including healthcare, finance, and even social media apps and search engines. However, along with the increased sophistication and power of these systems, the techniques of attacking them also increase. Among the emerging threats is the abuse of Unicode-based text masking, wherein attackers abuse textual input by using emojis, zero-width characters, and homoglyph replacement to bypass AI-powered content filtering and moderation systems. Due to the recent introduction of adversarial prompt injection against large language models (LLMs), attackers now have a way to bypass input validation systems and cause potential security risks, misinformation, and policy violation. This survey explores the territory of emoji and Unicode-based attacks on AI systems, focusing on recent developments, real-world use cases, and the existence of an imminent threat of such attacks that requires efficient defence measures.

AI technologies, particularly AI-based on NLP (Natural Language Processing) have taken a significant role in the processes of content moderation, cybersecurity, and information filtering. Whether it is spam detection and hate speech or malicious input filtering in conversational agents, AI is increasingly being utilized to maintain safety on the Internet. This however has also generated an increase in the level of advanced evasion techniques being employed by the criminals to bypass these systems.

One of these new ways of action has become the masking of data with emojis, Unicode characters and invisible symbols; a strong, yet low-key form of manipulation. Malicious actors can evade AI-based keyword detectors by incorporating harmless-looking emojis or zero-width characters, tokenization schemes can be obfuscated, and in some cases even result in AI misclassifications or hallucinations.

Even though Unicode-based masking attacks on AI systems are a serious issue, there isn't much systematic research on the subject. The purpose of this survey paper is to present a systematic review of the new threat landscape. We categorize and explain various Unicode-based obfuscation methods, evaluate their compatibility with AI models, present actual Unicode abuse cases, and look at the latest defences. Furthermore, we highlight the significant open problems and offer promising directions for further study to improve AI models' resistance to Unicode-based adversarial attacks.

II. BACKGROUND AND RELATED WORK

Attacks have been a thing in cybersecurity for some time now and have been employed in phishing attacks, domain spoofing, and malware distribution over the years. Their application to AI systems, in particular to those ones using transformer-based architectures is new.

The most popular direction of adversarial machine learning research has been perturbations of word or pixel embeddings [4]. However, with the introduction of Unicode manipulation and emojis and non- visible characters, NLP tokenizers and prompt integrity are new vulnerabilities. Some of the earliest academic works focused specifically on Unicode exploitation in AI applications would be pieces like those of X et al. [4] about timely injection.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue VI June 2025- Available at www.ijraset.com

Manipulation of digital content by using Unicode characters is not a recent invention, but it is only now that it has been actively applied against artificial intelligence (AI) systems, primarily large language models (LLMs) and AI-based filters. To understand the mechanism of such masking attacks, it is necessary to get acquainted with the corresponding concepts related to Unicode encoding, text processing with the AI, and adversarial attacks on machine learning systems.

A. Unicode and Emoji Encoding

Unicode is a standardized character encoding system designed to map text in computers worldwide, consisting of scripts of different languages and special symbols and emojis. Emojis in particular are depicted by sequences of Unicode code points and, often, combine special zero-width joiners (ZWJs) to create composite characters. Such expressiveness makes Unicode quite powerful and, at the same time, prone to misuse. Hackers can insert emojis or zero-width characters on otherwise innocent-looking strings of text and it will confuse or slip through the content during processing by AI models.

B. Data Masking and Obfuscation in Cybersecurity

The terminology of data masking is widely used to refer to the hiding or alteration of data in order to protect sensitive data. Unfriendly conditions utilize data obfuscation, which adds confusion or avoidance expertise to detection mechanisms. Homoglyph attacks (the replacement of visually similar characters of various scripts by one another) are a factor of many phishing and URL attacks. However, the most recent one is the demonstratively tokenization vulnerability of NLP tools and LLMs via emojis and Unicode idiosyncrasies.

C. AI Models and Vulnerability to Unicode Manipulation

Transformer-based models, including OpenAI's GPT series and Google's PaLM models, rely on Natural Language Processing (NLP) models to tokenize text to construct meaningful units. Unicode attacks imply that the vulnerabilities of tokenization approaches are either to break keywords using emojis or to insert invisible tokens that disorient the AI understanding. It renders systems of content moderation, spam filtering, or real-time directing avoidable, as shown in research such as Emoji Jailbreaks by Google Cloud [1] or the technical reports by Repello.ai [3].

D. Prompt Injection and Adversarial Attacks on AI

Prompt injection is an adverse attack of a specific type, in which the malicious input is crafted to manipulate the behaviour of an AI model. The recent form of this attack is called emoji-injection, where harmless-appearing characters are used to insert unlawful commands into LLM prompts [3].

One of the first academic resources in the formal study of this attack is the work of X et al. on Prompt Injection Attacks Using Unicode Manipulation [4].

III. CLASSIFICATION OF EMOJI AND UNICODE-BASED ATTACKS

The following section categorizes the primary types of Unicode-based data masking attacks currently observed in AI system exploitation. Each category highlights how these techniques evade AI processing, cause misclassification, or enable prompt injection.

A. Emoji Insertion Attacks

Emoji insertion attacks involve placing emojis before or within keywords in a way that they cannot be caught and as a way of confusing AI-based moderation. Since in the majority of AI tokenizers, emojis are regarded as individual tokens, including emojis between the offensive words may prevent keyword matching altogether. An ill-minded user wishing to post "violent material" may post it as v🚀iolent to bypass the filtering, by including the rocket emoji character as a token-breaker within the expected token-string.

Such an approach is increasingly used to inject prompts on large language models (LLMs) in order to bypass constrained instructions or introduce undesirable behaviour by the attackers. The evaluation of Emoji Jailbreaks by Google Cloud demonstrates the feasibility of how the arrangement of emoji can be used to manipulate LLM prompts even without breaking free of trivial input sanitization [1].

B. Zero-Width Character Exploitation

Zero-width characters such as the zero-width space (U+200B) or zero-width joiner (U+200D) do not render in final text but can break string processing in AI pipelines. By placing these characters inside important words, the attackers are able to render the content meaningless to models which rely on solely visual or lexical tokenization. Example The password string (zero-width space character between "pa" and "ssword") appears normal but can occasionally bypass keyword-based detection filters, especially in regular-expression based systems.

C. Homoglyph Substitution Attacks

Homoglyph attacks are based on visually similar or even identical looking characters of other scripts (e.g. Cyrillic a and Latin a). Homoglyphs are most often used in phishing URLs, but they are recently gaining popularity as a means to attack NLP-based systems as well. The replacements are not detected by AI, especially when the tokenizer of the AI model or the Unicode normalization algorithm fails to deal with script boundaries correctly.

Example: Writing “password” (using Cyrillic “p”) instead of “password” might defeat filters that assume input is entirely Latin-script based.

D. Combining Marks Manipulation

Diacritical marks may be combined with ordinary characters, visually corrupting them or rendering them unreadable by some text parsers. Although their application is mostly limited to aesthetic purposes, attackers can also use them to poison tokens and cause NLP pipeline failure.

Example: “Á b̈ c̉” adds combining marks which can cause input interpretation distortion on some models.

E. Mixed Script and Encoding Attacks

Confusable sequences combine several Unicode tricks to make inputs that are human-readable but should confuse AI, including emojis, homoglyphs, and zero-width characters. Such composite plans have a higher chance of getting through advanced AI-based filters.

IV. CASE STUDIES AND REAL-WORLD EXAMPLES

Unicode manipulation attacks on AI systems are a newer idea, but a number of high-profile incidents in the real world have proven that the concept works. These instances demonstrate the practical use of Unicode-based masking, against both cybersecurity systems, as well as advanced AI models. misclassification, or enable prompt injection.

A. Emoji Jailbreaks in Large Language Models

The Google Cloud researchers showed one of the first widely recognized examples of Unicode manipulation to be used in LLM exploitation [1]. The researchers found that emojis or other Unicode characters could be injected into chains of prompts to circumvent layers of moderation, and therefore injection of prompts could be achieved even in fully safe deployments of LLM. Attackers carefully placed emojis between forbidden tokens, which caused the LLM to overlook security policies or produce unauthorized texts. To give an example, automated keyword-based input blocking would be evaded by attackers typing a blocked keyword, e.g., delete □logs, which would inject malicious commands into the context visible to the model.

B. Emoji Obfuscation in Social Engineering Campaigns

Besides LLMs, Unicode masking has been widely employed in phish mail, spam and malware delivery, though mostly as part of social engineering attacks. A report by The Economic Times [2] has noted that cybersecurity experts have noted that the attackers are using emojis in spam email to get past spam filters besides AI-based phishing filters. At least once in a documented instance, the attackers have encoded apparently random emojis into domain names, and hyperlinks generating malicious links that looked benign to the average viewer and even some automated detection programs

C. Practical Prompt Injection Using Emojis

Repello.ai [3] detailed a comprehensive study of practical prompt injection, in which adversarial inputs were found to utilize Unicode idiosyncrasies in order to execute illicit code chunks or to escape conversational model restrictions.

These tests demonstrate that the apparently harmless symbols like hearts, stars, or objects can be utilized in opposition to language model safeguards and make emoji-based injection an efficient instrument of attackers.

D. Unicode Exploitation in Academic Research

A recent study by X et al. [4] represents an elaborate description of the Unicode exploitation towards adversarial machine learning. They document in their study that zero-width characters as well as homoglyph substitutions could both generate malformed inputs that could mislead transformer-based models during text parsing, posing dire consequences to the AI implementation in various industries. These case studies show that Unicode-based masking, which was so far associated only with web attacks and phishing, is now an immediate and obvious threat to the integrity of AI and requires focused research and effective mitigation measures.

V. CHALLENGES IN MITIGATING UNICODE-BASED ATTACKS

Although Unicode-based data masking attacks on AI systems are becoming increasingly complicated, there are a few technical and operational difficulties in defending against them. The principal reasons behind these are the inherent complexity of Unicode encoding, the variety of AI tokenization schemes, and the complicated state-of-the-art adversarial attack techniques.

A. Limitations in Tokenization Algorithms

The tokenization scheme (WordPiece or byte pair encoding, BPE) is particularly vulnerable to transformer-based language models and AI models in general. Since tokenizers are usually optimized for efficiency and language coverage, they lack adversarial robustness awareness. Unicode characters such as emojis, zero-width joiners, and combining marks can cause confusion for tokenizers by:

- Eliminating crucial symbols.
- Generating token combinations never seen before.
- Sequences that don't make use of preset filters.

Such malicious obfuscations cannot be systematically recognized or normalized by the tokenization algorithms currently in use [4].

B. Variability in Unicode Normalization Standards

Different platforms and programming frameworks have different approaches to Unicode normalization. AI pipelines have the freedom to interpret Unicode Consortium standards, unlike forms like NFC (Normalization Form C) and NFD (Normalization Form D), which are defined by those standards. This discrepancy makes coordinated defence much more challenging by encouraging attackers to create inputs that will behave differently across systems.

C. Evasion of Traditional Security Measures

Unicode masking attacks cannot be secured via the traditional security controls such as heuristic analysis, signature-based detection, or regular expression-based filters. These techniques can easily be defeated by invisible Unicode characters or script-switching homoglyphs since many of these functions operate on visible strings, or strings that are semantically interpreted [2].

D. Escalation Complexity of Attack Patterns

Also, attackers are increasingly applying Unicode masking in addition to other offensive methods like context-dependent prompt engineering and multi-stage injection. Such hierarchical complexity enables it to be difficult to identify all such instances using automated AI content moderation systems unless they include advanced contextual reasoning or machine learning-based adversarial defences [3].

VI. MITIGATION STRATEGIES

The protection of AI models against Unicode data masking attack is in the form of a layered defense, including preprocessing, adaptive detection, and strategic model alignment. Some of most notable strategies that are being investigated/ being implemented include the following:

A. Unicode Normalization in Preprocessing Pipelines

The first line of defence is the normal use of Unicode normalization standards in preprocessing of input. Fold visually equivalent Unicode sequences into their canonical form using normalization forms such as NFC (Normalization Form C) or NFKC (Normalization Form KC) so that they are no longer susceptible to homoglyphs or combining marks. Normalization should be a part of AI pipelines both in input and intermediate representations.

B. Invisible Character Detection and Removal

Surreptitious manipulation can be avoided by using special filters to identify and remove invisible characters such as zero-width spaces (U+200B), zero-width joiners (U+200D) and zero-width non-joiners (U+200C). Unicode-smart regex patterns or parsers can be deployed to warn or clean inputs before they reach downstream AI processing units.

Example Regex for detecting zero-width spaces: [\u200B\u200C\u200D]

C. Homoglyph Detection Algorithms

The detection of homoglyphs libraries (Homoglyphs or International Components for Unicode (ICU) tools) may help to detect and mark suspicious text as script mixing is used. Such exploitation of cross-script character similarity can be prevented by the addition of visual-similarity analysis or confusable mapping.

D. Contextual Input Sanitization for Prompt Injection

When it comes to language models in particular, the prompt injection risk may be overcome through the use of context-sensitive sanitization controls by:

- Detecting unusual Unicode sequences.
- Structure validation of inputs.
- Blocking or filtering the inputs with known adversarial Unicode strings [3].

In addition, instruction grounding and prompt templating in combination can diminish the impact of adversarial prompts that make it past the first line of defence.

E. AI-Based Anomaly Detection Systems

A more flexible secondary line of defence is using anomaly detection systems trained on machine learning to examine unusual Unicode patterns. Such systems are able to analyse not just individual characters, but input patterns, token distribution and context changes that are common to malicious manipulation.

F. User Awareness and Policy Integration

When AI systems process user-generated content, user training continues to be a top priority.

Platforms need to:

- Offer instructions for valid input forms.
- Users should be discouraged from manipulating Unicode.
- Including Unicode policies in the cybersecurity model.

VII. FUTURE RESEARCH DIRECTIONS

Due to the dynamic nature of Unicode-based masking attacks, research in this area must continue in order to remain into adversarial innovation. The important areas of further research are:

A. Development of Unicode-Robust Tokenizers

The study of Unicode-sensitive tokenizers algorithms is required to minimize the impact of adversarial inputs on NLP models. The new tokenizers should have a native support of script boundaries, invisible characters and homoglyph mappings without losing linguistic richness.

B. Standardization of Unicode Handling in AI Pipelines

Industries having agreed on Unicode preprocessing of AI inputs can enable cross-platform consistent defences. Such standards are to be formulated by coordination of AI research communities, members of Unicode Consortium and security communities.

C. Real-Time Detection Systems for Prompt Injection

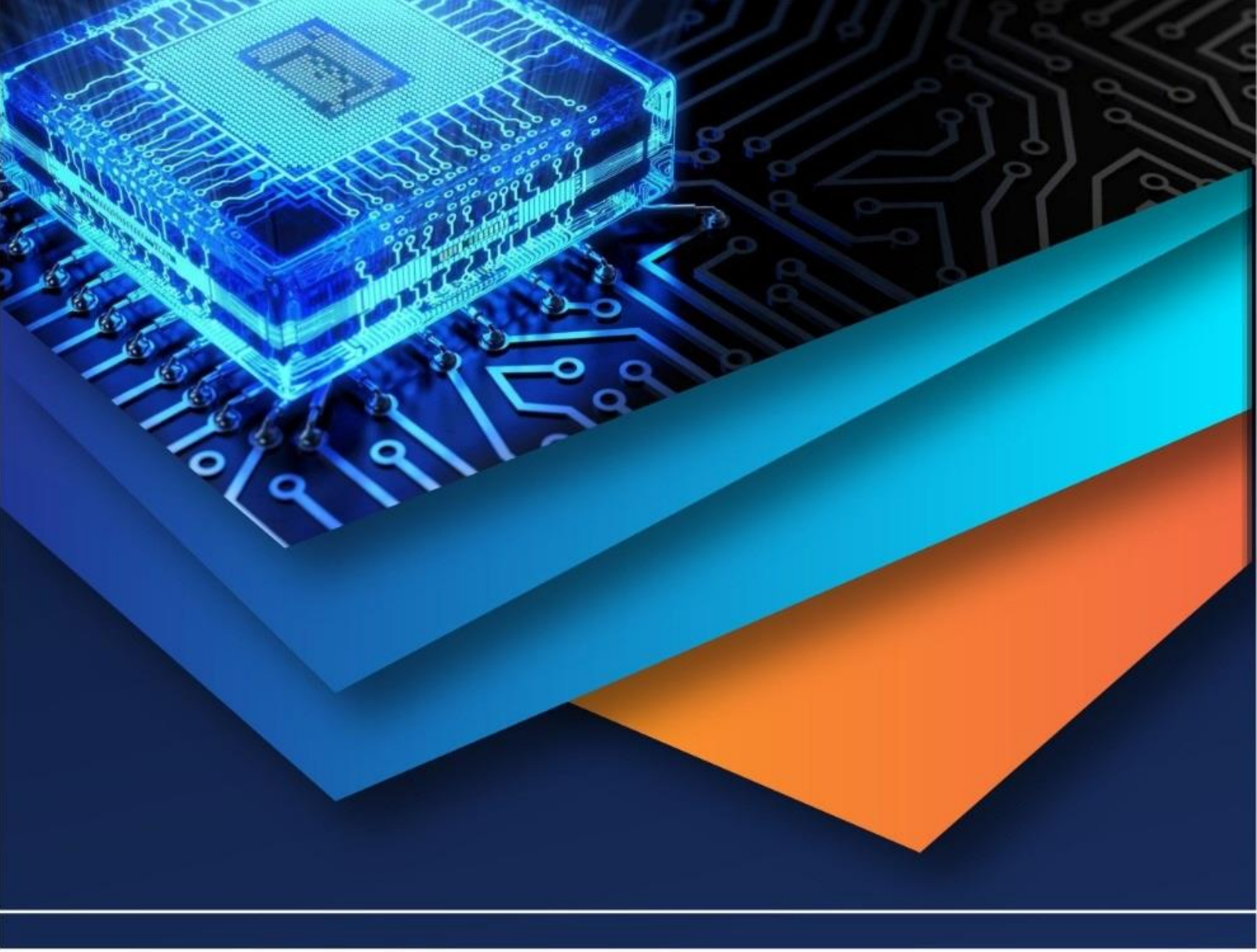
Designing real-time, low-latency detection systems specific to injection attacks on LLMs will be important, when interactive AI systems are deployed in high-risk applications, such as in healthcare or finance.

D. Large-Scale Dataset Creation for Unicode Attack Research

Open datasets of adversarial Unicode attacks on AI systems do not currently exist. The upcoming work needs to be done to compile and distribute such datasets to develop supervised learning solutions to Unicode threats.

REFERENCES

[1] Google Cloud, “Emoji Jailbreaks: Breaking AI Models with Unicode,” Medium, 2024. [Online]. Available: https://medium.com/google-cloud/emoji-jailbreaks-b3b5b295f38b
[2] “How emojis are becoming AI's weakest link in cybersecurity,” The Economic Times, 2024. [Online]. Available: https://economictimes.indiatimes.com/magazines/panache/how-emojis-are-becoming-ais-weakest-link-in-cybersecurity/articleshow/120253502.cms
[3] Repello.ai, “Prompt Injection Using Emojis,” 2024. [Online]. Available: https://repello.ai/blog/prompt-injection-using-emojis
[4] X et al., “Prompt Injection Attacks Using Unicode Manipulation,” arXiv, 2024. [Online]. Available: https://arxiv.org/pdf/2411.01077
[5] Unicode Consortium, “Unicode Technical Standard #39: Unicode Security Mechanisms,” 2023. [Online]. Available: https://www.unicode.org/reports/tr39/
[6] N. Carlini, A. Mishra, et al., “Poisoning Web-Scale Training Datasets Is Practical,” arXiv, 2023. [Online]. Available: https://arxiv.org/abs/2306.04634
[7] J. Jia and P. Liang, “Adversarial Examples for Evaluating Reading Comprehension Systems,” in Proc. of EMNLP, 2017, pp. 2021-2031.
[8] M. Zhang et al., “Adversarial Attack and Defence on AI-Based Content Moderation Systems,” IEEE Access, vol. 10, pp. 70312–70324, 2022.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY