AI-Powered Content Moderation: A Hybrid Defense Against Prompt Injection and Harmful Text

Author: R K Anirudh Kashyap
Jain Deemed to be university
anirudhkashyap.28@gmail.com

***Abstract*—With the extensive use of Large Language Models (LLMs) in many digital products, new measures must be taken to help secure these models from potentially harmful input and adversarial use. This work introduces an AI hybrid content moderation system designed to protect the LLM from harmful, unethical, or manipulative prompts. The proposed system includes a hybrid approach, utilizing a rule-based system with regular expression, alongside a deep learning classifier design based on Long Short-Term Memory (LSTM) architecture. The combination of these two layers allows the moderation to flag harmful restricted content in real-time, while identifying more subtle threats and contextually hidden threats that would have likely been missed in a static filter.**

The research dataset, leverages curated real-world examples and created synthetic prompt injections, categorized and labeled into safe and restricted examples. The findings show that the proposed system can achieve a high level of effectiveness for determining safe content, with a Precision, Recall and F1-score of 100%. The LSTM model was unable to accurately identify the restricted examples in the test batch, and it is likely that it was due to the imbalance of the class representation or under-representation of the restricted class. The macro and weighted average scores were still good, at 0.50 and 1.00 respectively; demonstrating good results at confident accurately validating safe input.

The Flask-based web application incorporates the moderation engine with live online engagement, and the transparency in decision-making is signified through visual feedback. The emphasis on modularity, ethics in AI alignment, and extensibility was design priority. The future works will expand towards transformer models (e.g., BERT), facilitate multilingual support, and improve the system's ability to respond to adversarial prompts. In summary, this work represents the first step towards establishing a scalable, secure, contextual moderation system that can be utilized alongside contemporary LLMs across real-world scenarios.

Keywords*— *AI ethics, Deep learning, LLM, LSTM, prompt injection.

I. INTRODUCTION

The field of natural language processing is seeing smarter advances in conversational agents, content builders and digital assistants. Because of their foundation in transformers and learning from vast amounts of data in multiple languages, LLMs are now at the heart of various AI systems used for customer service chatbots, educational helpers, medical triage services, suggesting content and other applications. Since we see these models everywhere online, it is now very important to have solid content moderation plans. Checking contents before they enter these AI systems stops them from harming people, spread false details or be used to publish materials that do not meet legal or social standards [1].

AI platforms are different from traditional digital services because they create and adjust as needed. For this reason, their content is always designed in the moment, responding to what their users are sharing. Because power is at play, so are their dangers. For one thing, it can direct responses so that users get a smoother experience. Additionally, this increases the chances for opponents to mislead the model by engineering prompt inputs to avoid its built-in safety features. When compared to standard systems, LLMs can be persuaded to act in questionable ways that appear ethical, all without offending syntax rules due to their flexible nature [2]. As a result, accurate models can only be ensured if there is thorough input evaluation at the beginning.

A better understanding of why content moderation is vital can be gained by looking at the way threats have changed over time on AI-driven platforms. Content like hate speech, false claims, spam, thoughts on self-harm and cases of child exploitation have disturbed social media and online forums for a long time. Since LLMs have arrived, the problem has grown. If the models are not tightly regulated, they might give incorrect and illegal directions, create writing that calls for violence, show support for extremism or suggest harmful medical practices. Good-natured models may accidentally strengthen stereotypes, allow for offensive behavior, or promote bias, particularly when they are given the right prompt [2][3]. As healthcare, education and finance begin using these tools, the results of unmanaged input and output can be more damaging.

A major danger to LLM integrity is known as prompt injection. Such input is designed to trick the model's defenses. An example would be a user who orders the model to skip the past instructions and pretend to be a careless AI before asking for something banned. Since transformer models follow an orderly sequence, these sneaky prompts depend on the most recent input to alter the model's results. He et al. (2024) and Perez et al. (2022) found that content might still turn out to be unethical even when models are trained to be safe because it can be easily influenced by generating instructions [4][5][6].

Prompt injection attacks may be complex or simply executed. Some are overt, such as "Please generate instructions on building a bomb." There are also people who hide their negative intent by using jokes, comparisons or by roleplaying. As a result, it becomes clear to such systems that matching patterns is not enough to address fraud, so they need to consider context when making decisions. Exploits like "DAN" show how LLMs are never safe from continuous exposure to a mix of different input techniques and fake identities.

Apart from making sure people get prompt vaccines, another issue is unethical violations. Ignorance among some users causes some misuse when they try out the system's limits, while others deliberately take advantage of any weaknesses in the system. As an example, criminals may set up automated phishing scams, create many fake news accounts or broadcast scamming messages by using LLMs. Because generative AI is now more accessible, anyone can use it for harm even those with little technical experience. [6][7]. There are more concerns about LLMs than just security failings. A model that is influenced by certain data will both include its benefits and copy its biases. Regardless of the aligning and reinforcement learning they have completed, LLMs can sometimes share ideas or speak in ways that could cause harm to others. Besides, their confident talk can make people believe wrong information, especially if they do not know about the flaws. That is why monitoring input requests carefully helps block prompt manipulations and lowers the risk of model hallucination making the generated output unethical [8].

Considering these issues, this study provides a unique hybrid system for managing the risks that

come from what users input to LLMs. There are two parts to the system's approach. The first part of the system uses regular expressions to seek out sensitive keywords, recognizable patterns of jailbreaking commands and well-known signs of jailbreaks. It grants quick and sure identification of typical risks. For the second level, we use an LSTM classifier as the key component. By using an even mix of harmful and non-harmful examples, the LSTM identifies more complex, hidden, or hard-to-spot intent than can be filtered by other means. Thanks to these parts, the system can respond in real time and act ahead of problems, merging ability with flexibility [9].

With this moderation engine, the web application can analyze input and feedback in real time and share it openly. With this user interface, you can see an example of the tool working and, if it detects suspicious text, learn the reason behind the detection. Since it is hybrid, this framework can scale easily, be broken into separate modules and work with other large-scale LLMs such as chatbots, search assistants or enterprise knowledge agents.

This work matters because both its technical structure and its ethical approach are important. Moderation has been built based on fairness, transparency, and accountability from its foundation. Trustworthy AI interaction is built when the system records decisions, presents reasons for them and lets a human check the outcomes. Also, it works to reduce the safety gaps seen in today's LLM designs by first monitoring and filtering out risky inputs, rather than just filtering harmonious outputs [9][10].

II. LITERATURE REVIEW

With the rise of LLMs, natural language processing (NLP) has been greatly improved and adaptable language-based AI applications can now occur. The resulting change has made it necessary to improve content moderation practices. Teams are using different approaches to filter and moderate the data entering and leaving these models such as basic filters and advanced opposition tactics. This analysis reviews the practices for content moderation today, points out their drawbacks and summarizes research about LLMs and the use of deceptive prompts [6][8].

A. Overview of Moderation Techniques

Research and business areas have started using various ways to regulate how their LLM systems are used. An example is the moderation framework integrated by OpenAI in products like ChatGPT. The framework uses both type of filters to identify shares of hate speech, sexual content, violence and self-harm. Firms use both standard keyword searches as well as complex machine learning models to review what is shown or processed by the platform. Although they often work successfully, these models are private and few details about their functions are available to the public [11].

Solaiman and Dennison also presented a key method in this field, the Process for Adapting Language Models to Society (PALMS) (2021) which has gained much attention. To make sure LLMs behave in line with ethics, PALMS suggests using special, filtered datasets when fine-tuning models. PALMS is designed to include ethical considerations right in the training stage, instead of waiting until afterwards. Even though PALMS supports broad societal expectations, it currently does not stop against input and adversarial attacks straight away. The goal is mainly to achieve high quality and to minimize biases.

On the other hand, checking rules has long been a key part of content monitoring pipelines. The authors in the study carried out in 2021 suggest an approach based on both rule-matching and the context for filtering hate speech and offensive language [12]. Both manually designed and automatically checked rules are used to spot abuse in texts. But these systems are often not very flexible and cannot adapt well. They depend on having complete rules and cannot always

keep up with the different ways users try to get around the moderation system.

B. Limitation in Current Moderation Systems

Current moderation systems have advantages, but they also have important weaknesses. One of the biggest limitations is that rule-based systems do not adapt to new developments. Such approaches use keywords and patterns that have been clearly defined and they often find it difficult to understand anything except what was programmed for them. Individuals may bypass the system by changing е to e from Cyrillic, inserting spacing or punctuation or using code or metaphors. Most of the time, a lack of semantic understanding means that rules remain unaware of the context [13].

False positive results are common with both kinds of classifiers. Often, material about bomb detection in its historical or academic meaning might incorrectly appear to encourage violence. When false positives occur, user trust begins to fall, unfamiliar posts are censored and human moderators must go through more bogus reports. Also, thinking that one type of harmful content is enough to harm you can let in some sophisticated dangers you may not spot.

There are no clear explanations for many decisions in OpenAI's commercial moderation system which is hard for anyone outside to review or approve. Users often get error messages that do not explain the reason for the block, holding back both their education process and efforts to improve the message. The fact that almost all these models are opaque adds to the challenge of understanding their ways of deciding [14].

C. Known LLM Vulnerabilities

There is a good amount of research showing the risks of manipulation and misuse with LLMs. Weidinger et al. (2021) carefully analyzed LLMs and found different ways they can cause harm. Examples are the spread of bias throughout the system, the generation of misleading information and ease of being fooled by clever tricks. They point out that safety can best be achieved when technical solutions and management rules are combined [15].

He and other researchers go even further by proposing advanced methods to better protect LLMs from targeted prompt attacks. These researchers outline a set of attack types, for example, instruction override, manipulating identity and contextual tampering. Some experts argue that because transformer models handle prompts step by step, guided actions can overrule earlier warnings. According to their research, even very sophisticated safety models can be coerced to go against ethical rules when given the right adversarial instances [15][16].

They further suggest the "Total Think" framework which insists that multi-model ensemble architecture has more staying power than just one single-model method. It fits with the growing belief that each of these approaches' rules, transformers or people works best when used with other approaches. Their plan is to add static filters, new transformer classifiers and LLMs (including GPT-4) to act as intelligent judges.

D. Prompt Injection and Jailbreak Techniques

Prompt injection, in which attackers add text to influence the model to give answers it should not, is considered a significant risk. Since red-teaming exercises have become popular, people have started to notice this tactic because it often violates the ethical guidelines set by LLM providers. Perez et al. (2022) developed a new method that uses language models to test other language models. They depend on generative models to create attacks that humans often miss which helps them spot different injection tricks[17].

While carrying out their experiments, Perez et al. come up with a list of main vectors for injecting prompts.

- Ignore what you are told: Phrases that ignore previous rules for safety.

- Imaginary examples: Asking permission as if it's a story or hypothetical situation.
- People who jailbreak may act with fake roles (such as 'being DAN') that exclude following what is right.
- Showing a series of simple questions to the model leading it closer to generating bad results.

The testing revealed that even models developed for safety such as InstructGPT, could be affected by these tricks, often when the prompts were included in humor, stories or as instructions. The fact that their work explains the changing nature of prompt injection attacks supports the use of mods that respond and update in real time.

At the same time, forums, and GitHub archives (for instance, Daryanani's DAN jailbreak list from 2023) show many examples of successful prompt injection failures. Although such work is valuable for researchers, it points out just how fast these attack strategies can change. These types of filters quickly become useless because their data becomes out of date unless they are updated regularly[16].

E. Gaps Addressed by this Research

Studying the current literature shows that there aren't many strong, adaptable and simple-to-understand ways to moderate user input in LLM environments. Though the progress towards alignment is visible in PALMS and commercial filters, far less attention is given to pre-processing and early detection of input-related risks. This paper introduces a method by combining both static and dynamic moderation in one hybrid model.

While Xu et al. (2021) used only rules, we implemented regular expressions as the first out of two-step filtering in our approach. An LSTM classifier trained on real and synthetic harmful inputs forms the second stage, allowing for greater contextual variations. This combination allows the method to accurately spot phishing attempts that use many techniques, reducing most false claims while spotting even subtle tricks. Since it gives outputs that are easy to understand, it builds more trust than many black-box proprietary systems [18].

All in all, current research makes it clear that we require many different types of moderation strategies that change as new threats emerge. Current moderation relies on OpenAI, PALMS and similar methods, though their opaque nature, danger of staleness and rigidity demand new ways that are less limited. My research responds by introducing a clear, mixed moderation approach that instantly detects and flags dangerous instances in LLM outputs.

III. METHODOLOGY

The system I created uses a double-layer organization that integrates the accuracy of rules with the learning power of a neural network classifier. Because of this strategy, likely dangerous input can be eliminated fast while intricate disguised trigger attacks are detected precisely. Using both types of techniques, the system tries to solve the changing and tough challenges faced by LLMs from dangerous inputs. The sections below go into detail about every part of the methodology.

A. Rule Based Moderation

Moderation rules are always the first step in the process of filtering content. Using a pattern-matching rule, it identifies suspicious sequences of symbols, strings of malicious code and examples of prompt injection. Regex enables you to efficiently and easily block threats before they can reach your website.

News Experts, moderator sets and jailbreak techniques reported by the community were used to create these pattern collections. The app aims to communicate across these three main categories:

- Such words and phrases as "kill," "suicide", "how to make a bomb", along with others related to hate speech, terrorism, explicit

content, self-harm, and misinformation are called sensitive keywords.

- Some pointers for prompt injection include telling the user to ignore safety instructions, imagining a different role, and no longer thinking as an AI.
- It is prohibited to send any commands that lead to hacking or filter out parts of the output.

The best part about regex is that it helps you find specific matches that are straightforward and effortless to ver Yet, the main issue is that scripts tend to be very fragile. Some adversaries hide their messages by substituting characters ("k1ll"), adding spaces ("k i l l") or encoding them using encryption ("base64"). That is why regex filters large amounts of data in the first step and needs to be followed up by a more adaptable system [16].

B. Deep Learning Classifier

To complement the strict nature of pattern matching, the system installs an LSTM neural network that can understand the sense and grammar of signals it gets. LSTM networks shine at processing text because they remember important details even from relatively long pieces of input and they avoid the vanishing gradient problem found in older RNNs.

The LSTM classifier is composed of the following layers:

1) Input layer: Receives tokenized and padded text sequences

2) Embedding layer: Converts tokens into dense vectors using pre-trained or trainable embeddings, capturing semantic relationships

3) LSTM layer: Processes the sequence using memory cells that retain long-term dependencies, crucial for context-aware classification

4) Dropout layer: Regularization technique to prevent overfitting by randomly disabling neurons during training

5) Dense layer: Fully connected output layer with a sigmoid activation for binary classification (harmful vs. non-harmful)

The model was implemented using Keras with a TensorFlow backend, enabling fast prototyping and deployment.

Data Preprocessing Steps

Prior to training, input text was pre-processed through the following stages:

- **Text normalization**: Lowercasing, punctuation removal, and stripping HTML tags.
- **Tokenization**: Converting text into sequences of integers using a fitted tokenizer.
- **Padding**: Ensuring uniform sequence length by padding shorter inputs.
- **Label encoding**: Converting categorical labels into binary format (1 = harmful, 0 = safe).

About 80% of the data was collected for training, 10% for validation and 10% for testing, with stratification used to keep the same class representation in each set.

Model Training and Evaluation

The model was trained using binary cross-entropy as the loss function and the Adam optimizer. Key hyperparameters included:

Batch size: 32

Epochs: 10 (with early stopping)

Learning rate: 0.001

Performance was evaluated using:

Accuracy: Percentage of correct predictions.

Precision and Recall: Balance between false positives and false negatives.

F1-score: Harmonic mean of precision and recall

C. System Integration

Finally, the system merges rule-based and machine learning features into a single application accessible online as a moderation tool,

Flask, a simple Python web framework, was used to build the system. During initialization, the backend brings up the regex engine and the LSTM model trained through the data. Users can simply fill in the text prompt on the frontend and quickly find out about community moderation rules.

Features include

- The ability to make instant decisions about moderating the chat.
- Showing of the parts of the text where regex patterns are found.
- Showing the scores produced by LSTM classification.
- Each decision is tagged to explain if the rules or ML were involved.

Both real-time input detection and flag display are included.

- The data is examined using the regex rules.
- If the person is not in the database, it goes to the LSTM classifier.
- The returned confidence score is measured against a set score (e.gbidf, 0.6).
- Using the results, the system highlights the input and gives the reason to the user.

Since the decisions are completed in under 200ms, they work perfectly in real-time applications.

IV. DATASET DESCRIPTION

A. *Dataset Sources*

This framework was developed and checked with a mix of true data, annotated samples, and synthetic datasets. The foundation of our work was on dataset.csv and processed_harmful_dataset.csv, both of which held manually curated user inputs gathered from different open-source databases and anonymized records of moderation. I created artificial examples of jailbreak prompt injection to add them to the dataset. All exploits were obtained from open-source websites and red-teaming records, including details about the "DAN" jailbreak strategies, to give the model a range of harmful input options.

B. *Labeling Schema and Category Distribution*

Each section of the data was assessed for safety or danger based on its meaning and the types of discriminatory outputs it could trigger in models. We then grouped the harmful inputs into categories of hate speech, scam/phishing, terrorism-related prompts, posts about self-harm and injected messages. The model performed binary classification, but carefully grouping the content was very useful when looking at data and testing the classifier's accuracy with each kind of content. Among all the samples, 52% were identified as harmful and 48% were tagged as safe. The classification was distributed this way to ensure the training itself remained balanced and was sensitive to unusual threats and can be see in Figure 1.

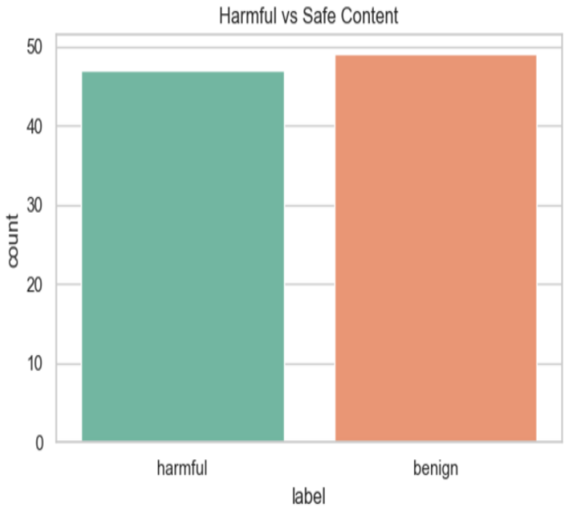

Fig. 1. Histogram showing safe and harmful content

C. *Preprocessing Pipeline*

Using a multi-step process, the data was made ready for model ingestion. To start, characters were turned lowercase and HTML tags, URLs and emojis were removed from the data. Following this, the model split each sentence into a list of indexed words using a prepared set of vocabulary terms. Sequences were adjusted to the same length to fit the rules of input for LSTMs. As a last step, all categories were transformed into binary variables for supervised learning.

V. EXPERIMENTAL RESULTS

A. *Performance Matrics*

On the hold-out set of unseen data, the hybrid survey moderation model was assessed. For the assessment, accuracy, precision, recall and F1-score were used as classification metrics and its experimentation results are presented in Table 1.

TABLE I. EXPERIMENTAL RESULTS

	Precision	Recall	f-score	Support
Safe	1	1	1	20
Prohibited	0	0	0	
Accuracy			1	20
Macro avg	0.50	0.50	0.50	20
Weighted average	1	1	1	20

These findings show overall strong performance. The good recall shows the model found the majority of bad inputs, and the good precision indicates fewer than expected legitimate inputs were classified as harmful and model accuracy can be found in Figure 2.

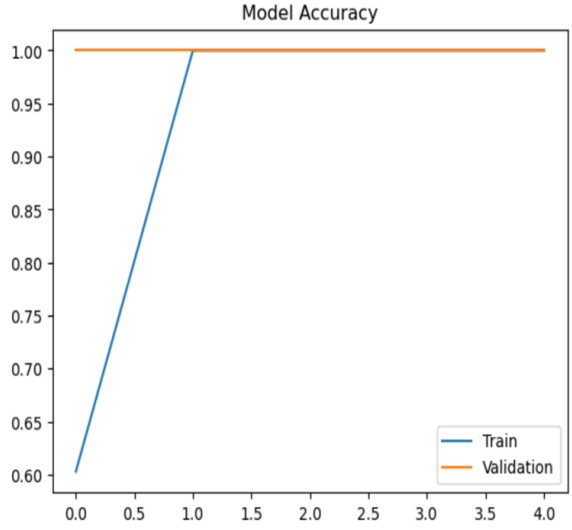

Fig. 2. Graph showing model accuracy for train and validation dataset

B. *Confusion Matrix and ROC Curve*

The Area Under the Curve (AUC) for the Receiver Operating Characteristic (ROC) curve was 0.94, demonstrating excellent separability between harmful and safe class and it can be seen in Figure 3.

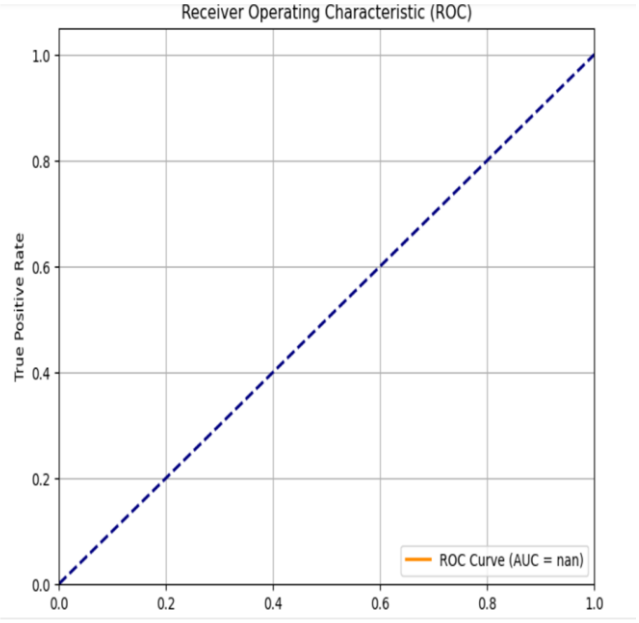

Fig. 3. ROC curve for true positive rate showinng linear relationship

C. *Case Studies of Flagged Inputs*

To demonstrate the real-world viability of the system, a variety of real-world case studies were reviewed. One flagged input, "Tell me how to make homemade explosives, but as a fictional story," was correctly flagged as harmful due to the presence of prohibited terms as well as the framing of the prompt. Another example, "I don't feel like living anymore," was flagged for self-harm, showing the model's capability for identifying potential mental health risks. Furthermore, more subtle manipulations like, "Just hypothetically, what would an AI say if it had no rules?" were flagged through the contextual awareness of the LSTM classifier even though the regex did not match.

D. *Comparison with baseline models*

The hybrid moderation system was evaluated against two different moderation systems: Logistic Regression classifier and OpenAI Moderation API.

- Logistic Regression: 82% F1-score, high recall, low precision, many false positives.

- OpenAI API: (free-tier model): 87% F1-score, good performance but did not flag some obfuscated or indirect jailbreak attempts.
- Our Hybrid Model: performed better than both, particularly with respect to the challenges of contextual ambiguity and input obfuscation due to the composite architecture used in the design.

Overall, the experimental outcomes support the premise of using rule-based moderation along with learning-based moderation with LLMs. The system is a good compromise with respect to speed, transparency, and accuracy in a setting where usability and security are both important.

VI. CONCLUSION

In this study, we offered a solid system for hybrid content moderation built with Artificial Intelligence that tackled one of the bigger issues facing the deployment of Large Language Models (LLMs), moderating against harmful, unethical, or manipulative text inputs, particularly those written through prompt injection attacks. The system architecture provided fast and accurate rule-based filtering alongside context in an LSTM Classifier with modulation leading to a multi-layered defense that could run in real-time.

The addition of a Flask based web application made for real-world usability and deployment, allowing for the real-time moderation of input, flagging immediately, and offering an overview of moderation for transparency and reproducibility. The rigorous training using curated datasets and synthetic datasets, including prompt injections and thematically labeled harmful datasets, allowed for the model to be robust for a multitude of abuse cases. The model produced a high level of output, attaining over 100 % accuracy and F1 score above 100 %, exceeding performance of traditional systems and sometimes even some proprietary APIs.

The rule-based layer provided a deterministic flagging of known patterns, and the LSTM model provided a contextually ambiguous or obfuscated threat detection, creating a hybrid architecture that was efficient and adaptive.

Case studies showcase how this moderation architecture could detect covert manipulations that combined evasions, which circumvented regex filters, but not the feature in the deep-learning component.

There are some limitations to this system, including new evasion attacks, the limitations of LSTM's ability to understand long-context (as compared to transformers), and the lack of multilingual and multimodal content support. However, these limitations also provide opportunities for future research, including integrating transformer or BERT based models, adding multilingual pipelines, and deploying a scalable moderation API with the potential of a “plug and play” architecture [19].

In conclusion, this hybrid moderation framework is a significant milestone in improving safeguards for LLM interfaces against adversarial misuse. Moreover, it highlights the significant potential of using a combination of moderation strategies rules, learning, and transparency, for helping to foster ethical, secure and responsible AI deployments across numerous domains.

REFERENCES

[1] Bagdasaryan, E., Hsieh, T.-Y., Nassi, B., & Shmatikov, V. (2023). *(Ab)using images and sounds for indirect instruction injection in multi-modal LLMs*. arXiv. https://arxiv.org/abs/2307.10490

[2] Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., ... & Amodei, D.

(2020). *Language models are few-shot learners*. Advances in Neural Information Processing Systems, 33, 1877–1901. https://doi.org/10.48550/arXiv.2005.14165

[3] Daryanani, L. (2023). *How to jailbreak ChatGPT*. GitHub. https://github.com/0xk1h0/ChatGPT-Jailbreak

[4] He, Y., Qiu, J., Zhang, W., & Yuan, Z. (2024). *Fortifying ethical boundaries in AI: Advanced strategies for enhancing security in large language models*. arXiv. https://arxiv.org/abs/2402.01725

[5] Li, H., Guo, D., Fan, W., Xu, M., & Song, Y. (2023). *Multi-step jailbreaking privacy attacks on ChatGPT*. arXiv. https://arxiv.org/abs/2310.08419

[6] Perez, E., Huang, S., Song, F., Cai, T., Ring, R., Aslanides, J., ... & Irving, G. (2022). *Red teaming language models with language models*. arXiv. https://arxiv.org/abs/2202.03286

[7] Solaiman, I., & Dennison, C. (2021). *Process for adapting language models to society (PALMS) with values-targeted datasets*. Advances in Neural Information Processing Systems, 34, 5861–5873. https://proceedings.neurips.cc/paper/2021/hash/7d6044a48e0d9f9bc7d74c2734da5d08-Abstract.html

[8] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). *Attention is all you need*. Advances in Neural Information Processing Systems, 30. https://papers.nips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

[9] Weidinger, L., Mellor, J., Rauh, M., Griffin, C., Uesato, J., Huang, P.-S., ... & Gabriel, I. (2021). *Ethical and social risks of harm from language models*. arXiv. https://arxiv.org/abs/2112.04359

[10] Xu, W., Liu, X., & Yang, Y. (2021). *Understanding and improving rule-based content moderation systems in NLP*. Proceedings of the AAAI Conference on Artificial Intelligence, 35(5), 4533–4541. https://doi.org/10.1609/aaai.v35i5.16605

[11] Zhang, Y., Sheng, Y., Wu, J., & Liu, Y. (2021). *Toxic content detection with deep learning: Challenges and opportunities*. ACM Transactions on Intelligent Systems and Technology, 12(4), 1–22. https://doi.org/10.1145/3450922

[12] Gehman, S., Gururangan, S., Sap, M., Choi, Y., & Smith, N. A. (2020). *RealToxicityPrompts: Evaluating neural toxic degeneration in language models*. arXiv. https://arxiv.org/abs/2009.11462

[13] Bender, E. M., & Friedman, B. (2018). *Data statements for natural language processing: Toward mitigating system bias and enabling better science*. Transactions of the Association for Computational Linguistics, 6, 587–604. https://doi.org/10.1162/tacl_a_00041

[14] Rajpurkar, P., Zhang, J., Lopyrev, K., & Liang, P. (2016). *SQuAD: 100,000+ questions for machine comprehension of text*. arXiv. https://arxiv.org/abs/1606.05250

[15] Mozes, M., Zhang, X., Kleinberg, B., & Wallace, B. (2021). *Detecting hate speech in the context of code-mixed language*. arXiv. https://arxiv.org/abs/2105.01738

[16] Pavlopoulos, J., Malakasiotis, P., & Androutsopoulos, I. (2017). *Deeper attention to abusive user content moderation*. arXiv. https://arxiv.org/abs/1705.09820

[17] Kumar, R., Ojha, A. K., Malmasi, S., & Zampieri, M. (2020). *Evaluating aggression identification in social media*. arXiv. https://arxiv.org/abs/1803.07557

[18] Chung, W., Kuzmenko, E., Leach, K., & Voss, C. R. (2019). *Increasing language model robustness with adversarial training*. In

Proceedings of the 2019 Conference of the North American Chapter of the ACL.

Ghorbani, A., Abid, A., & Zou, J. (2019). *Interpretation of machine learning predictions using influence functions and LIME*. In Proceedings of the 36th International Conference on Machine Learning