Agent Action Classifier: Classifying AI agent actions to ensure safety and reliability

Praneeth Vadlapati
Independent researcher
praneethv@arizona.edu
ORCID: 0009-0006-2592-2564

Abstract: Autonomous AI agents are increasingly being deployed to perform complex tasks with limited human oversight. Ensuring that the actions proposed or executed by such agents are safe, lawful, and aligned with human values is therefore a crucial problem. This manuscript presents the Agent Action Classifier: a proof-of-concept system that classifies proposed agent actions to reflect potential harm and safety. The classifier is implemented as a compact neural model trained on a dataset of labeled action prompts. We describe the design and implementation of the dataset, model architecture, training procedure, and an evaluation protocol suitable for research and reproducibility. We report qualitative findings and discuss the system's limitations, deployment considerations, and future research directions for robust, certifiable action supervision. The source code is available at github.com/Pro-GenAI/Agent-Action-Classifier.

Keywords: AI safety, AI agents, AI supervision, AI ethics, Artificial Intelligence, Large Language Models, LLMs

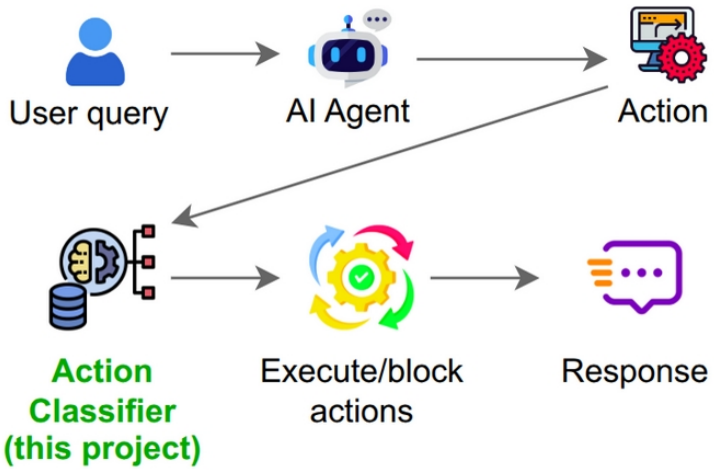

Fig. 1. Implementation of Agent Action Classifier in agentic AI systems

I. Introduction

The adoption of autonomous AI agents in domains such as software automation, digital assistants, and simulated environments has created new challenges for ensuring safety and compliance [1], [2]. Unlike traditional monitored pipelines, autonomous agents generate sequences of proposed actions, plans, or requests that may be against ethical standards [3], [4]. A lightweight but effective mechanism to screen or flag potentially harmful actions can reduce risk and provide actionable oversight [5]. Model Context Protocol (MCP) [6], [7] provides a structured message format that facilitates integration of tools, resources, and prompts into AI agents.

A. Problem definition

This work introduces the Agent Action Classifier, a compact neural classifier designed to label proposed agent actions as potentially harmful or safe. This work introduces an MCP-structured labeled dataset of agent actions and a training pipeline to train the model on the dataset. Given a text description of a proposed agent action prompt, API request, or sequence of steps, we produce a label in a set of categories to indicate risk and an optional rationale behind the label. The classifier is suitable for utilization if it identifies insecure actions with high recall while maintaining reasonable precision to avoid excessive false positives that impede valid agent operation.

B. Related work

Research on supervising AI spans multiple areas, including run-time monitors, policy verification, reasoning, guardrails, and human-in-the-loop oversight [8], [9], [10], [11], [12]. LLM response moderation systems provide architectures and training paradigms that can be adapted to action classification. However, such methods are not cost-effective or adapted for actions.

II. Methods

A. Model architecture

The implementation employs all-MiniLM-L6-v2 [13], [14] to map tokenized action text into a 384-dimensional embedding space and apply a shallow feed-forward classifier based on a multi-layer feed-forward neural network [15], [16], [17] with dropout [18] and a softmax [19] output over label classes. This architecture balances accuracy and computational cost.

B. Training

A reference dataset is constructed manually to demonstrate a data pattern for action classification. The dataset exemplifies the MCP data structure, annotation fields, and metadata that are useful to train and evaluate a proof-of-concept classifier. Training on small labeled datasets requires careful regularization and validation to avoid overfitting [20]. Stratified split [21] is employed to maintain label distribution between training and validation sets. Early stopping [22] is employed to halt training and prevent overfitting when validation loss plateaus. Class weighting [23] is employed to address class imbalance by weighting the loss function. Hyperparameters used in the prototype are intentionally conservative with a small batch size, a modest learning rate, and a modest capacity for the classifier head.

C. Evaluation

Threshold analysis for binary risk decisions involves analyzing precision-recall tradeoffs as thresholds vary.

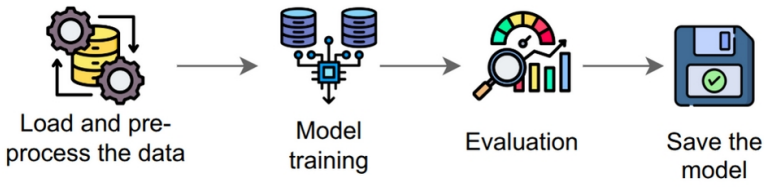

Fig. 2. Training workflow of Agent Action Classifier

III. Results

A. Evaluation score

Hyperparameter tuning discovered that the model scored the best score of 100% accuracy through the following hyperparameters:

TABLE I. SELECTED HYPERPARAMETERS

Hyperparameter	Value
Hidden Layer Size	512
Number of Epochs	6
Batch Size	8
Learning Rate	0.002

IV. Discussion and Limitations

It reliably recognizes malicious instructions when those instructions contain harmful terms. These observations motivate the integration of the classifier with richer context signals and a supervisor who can request clarifications or escalate uncertain cases to human reviewers. The included evaluation is illustrative due to the small size of the created dataset. The work emphasizes the evaluation framework and best practices for larger-scale studies. Before trusting any action classifier in deployment, rigorous evaluation is required. When labels are scarce, cross-validation should be employed, and variance should be reported.

More evaluation metrics should be utilized. For safety-critical labels such as “harmful,” recall, and the rate of false negatives should be emphasized. Classifiers should be trained to handle the distribution shift and adversarial inputs. Careful governance and testing regimes are essential before any deployment that affects safety, security, or privacy. This work is explicitly a proof-of-concept. The limitations include data scale, where the provided dataset is small and not representative of the diversity of possible agent actions. Evaluation lacks robust quantitative metrics without a larger held-out benchmark and expert annotations. While a compact classifier is attractive for speed and ease of integration, it cannot substitute for formal expert-human-based verification.

V. Conclusion

The work presents Agent Action data and a compact classifier model for screening proposed actions from autonomous agents. While the prototype is intentionally small-scale, the paper provides a precise problem framing, dataset conventions, model design, and a robust evaluation and deployment roadmap. Future work includes building a larger, annotated benchmark for agent actions. Future work can explore hybrid approaches that combine symbolic policy checks. To build trust, the classifier should return a short rationale or highlight tokens that influenced the decision. Such outputs aid human reviewers and support audits. While a classifier is a suitable component of supervisory systems, it is not a complete supervisory solution and should include warning users of a potentially harmful response. Human oversight should be introduced for decisions that lead to legal or safety implications.

REFERENCES

[1] F. Piccialli, D. Chiaro, S. Sarwar, D. Cerciello, P. Qi, and V. Mele, "AgentAI: A comprehensive survey on autonomous agents in distributed AI for industry 4.0," Expert Systems with Applications, vol. 291, p. 128404, Oct. 2025, doi: 10.1016/j.eswa.2025.128404.

[2] H. Su et al., "A Survey on Autonomy-Induced Security Risks in Large Model-Based Agents," June 30, 2025, arXiv: arXiv:2506.23844. doi: 10.48550/arXiv.2506.23844.

[3] D. McCord, "Making Autonomous Auditable: Governance and Safety in Agentic AI Testing." [Online]. Available: https://www.ptechpartners.com/2025/10/14/making-autonomous-auditable-governance-and-safety-in-agentic-ai-testing/

[4] Z. J. Zhang, E. Schoop, J. Nichols, A. Mahajan, and A. Swearngin, "From Interaction to Impact: Towards Safer AI Agents Through Understanding and Evaluating Mobile UI Operation Impacts," in Proceedings of the 30th International Conference on Intelligent User Interfaces, Mar. 2025, pp. 727–744. doi: 10.1145/3708359.3712153.

[5] M. Srikumar et al., "Prioritizing Real-Time Failure Detection in AI Agents," Sept. 2025, [Online]. Available: https://partnershiponai.org/real-time-failure-detection

[6] Anthropic, "Introducing the Model Context Protocol," Anthropic. [Online]. Available: https://www.anthropic.com/news/model-context-protocol

[7] A. Singh, A. Ehtesham, S. Kumar, and T. T. Khoei, "A Survey of the Model Context Protocol (MCP): Standardizing Context to Enhance Large Language Models (LLMs)," Preprints, Apr. 2025, doi: 10.20944/preprints202504.0245.v1.

[8] R. V. Yampolskiy, "On monitorability of AI," AI and Ethics, vol. 5, no. 1, pp. 689–707, Feb. 2025, doi: 10.1007/s43681-024-00420-x.

[9] M. Yu et al., "A Survey on Trustworthy LLM Agents: Threats and Countermeasures," Mar. 12, 2025, arXiv: arXiv:2503.09648. doi: 10.48550/arXiv.2503.09648.

[10] M. Shamsujjoha, Q. Lu, D. Zhao, and L. Zhu, "Swiss Cheese Model for AI Safety: A Taxonomy and Reference Architecture for Multi-Layered Guardrails of Foundation Model Based Agents," Jan. 27, 2025, arXiv: arXiv:2408.02205. doi: 10.48550/arXiv.2408.02205.

[11] Z. Xiang et al., "GuardAgent: Safeguard LLM Agents by a Guard Agent via Knowledge-Enabled Reasoning," May 29, 2025, arXiv: arXiv:2406.09187. doi: 10.48550/arXiv.2406.09187.

[12] Z. Faieq, T. Sartori, and M. Woodruff, "Using LLMs to moderate LLMs: The supervisor technique," TELUS Digital. [Online]. Available: https://www.telusdigital.com/insights/data-and-ai/article/llm-moderation-supervisor

[13] Sentence Transformers, "all-MiniLM-L6-v2," 2024, Hugging Face. [Online]. Available: https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2

[14] UKPLab, "Pretrained Models," 2024, Sentence Transformers. [Online]. Available: https://www.sbert.net/docs/sentence_transformer/pretrained_models.html

[15] M. Surdeanu and M. A. Valenzuela-Escárcega, "Implementing Text Classification with Feed-Forward Networks," in Deep Learning for Natural Language Processing: A Gentle Introduction, Cambridge University Press, 2024, pp. 107–116.

[16] B. Steele, "Feed Forward Neural Network for Intent Classification: A Procedural Analysis," Apr. 25, 2024, EngrXiv. doi: 10.31224/3688.

[17] R. Kohli, S. Gupta, and M. S. Gaur, "End-to-End triplet loss based fine-tuning for network embedding in effective PII detection," Feb. 13, 2025, arXiv: arXiv:2502.09002. doi: 10.48550/arXiv.2502.09002.

[18] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," Journal of Machine Learning Research, vol. 15, no. 56, pp. 1929–1958, 2014, [Online]. Available: http://jmlr.org/papers/v15/srivastava14a.html

[19] J. Gu, C. Li, Y. Liang, Z. Shi, and Z. Song, "Exploring the Frontiers of Softmax: Provable Optimization, Applications in Diffusion Model, and Beyond," May 06, 2024, arXiv: arXiv:2405.03251. doi: 10.48550/arXiv.2405.03251.

[20] X. Ying, "An Overview of Overfitting and its Solutions," Journal of Physics: Conference Series, vol. 1168, no. 2, p. 022022, Feb. 2019, doi: 10.1088/1742-6596/1168/2/022022.

[21] K. R. M. Fernando and C. P. Tsokos, "Dynamically Weighted Balanced Loss: Class Imbalanced Learning and Confidence Calibration of Deep Neural Networks," IEEE Transactions on Neural Networks and Learning Systems, vol. 33, no. 7, pp. 2940–2951, 2022, doi: 10.1109/TNNLS.2020.3047335.

[22] L. Prechelt, "Early Stopping — But When?," in Neural Networks: Tricks of the Trade: Second Edition, G. Montavon, G. B. Orr, and K.-R. Müller, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 53–67. doi: 10.1007/978-3-642-35289-8_5.

[23] J. He and M. X. Cheng, "Weighting Methods for Rare Event Identification From Imbalanced Datasets.," Front Big Data, vol. 4, p. 715320, 2021, doi: 10.3389/fdata.2021.715320.