ML to Predict Effectiveness of the MCP Authorization Model for LLM-Powered Agent

Upakar Bhatta
Information Technology and Management, Central Washington University, Ellensburg, WA, United States

***Abstract*—In today's AI-driven world, unlocking AI potential and enabling AI models to communicate with external data sources is vital for enhancing the efficiency and security of AI-driven applications. The Model Context Protocol (MCP) serves as a standard for maximizing AI potential. This study leverages a machine learning approach to predict the effectiveness of the MCP Authorization Model for an LLM-powered agent. It utilizes logs from Azure services such as Azure Monitor, Azure Sentinel, and Azure Active Directory, which are used to monitor MCP server activity, to create a sample dataset. This dataset includes features such as source_ip, destination_ip, event_type, alert_severity, and target_variable. These features are used to train the ML model to assess the effectiveness of the MCP Authorization model for LLM-powered agents, enabling organizations to better understand the importance of a secure connection between AI models. This approach contributes to unlocking AI's full potential while improving application security and operational efficiency.**

Keywords—Model Context Protocol; artificial intelligence; machine learning; large language model

I. Introduction

Large language models (LLMs) have evolved rapidly, demonstrating sophisticated reasoning and problem-solving capabilities [1, 2, 3]. In recent years, the rise of Large Language Model (LLM)-powered agents capable of interacting with various tools has gained significant momentum. Frameworks like LangChain [4] and LlamaIndex [5] facilitate the standardized tool interfaces, making it easier to integrate LLM-powered autonomous agents with external services.

These autonomous AI agents, operating in security-sensitive environments has necessitated appropriate security control policies to ensure compliant operations. In late 2024, Anthropic introduced the Model Context Protocol (MCP) for standardizing AI-tool interactions [6]. MCP specifies the rules on how external data sources and tools should interact with LLMs [7, 8] and provides a framework for AI applications to communicate dynamically with external tools. These models serve as a structured framework for enforcing security policies. However, their effectiveness in dynamic, AI-driven ecosystems remains an open challenge. This study explores the machine learning (ML) techniques to predict the effectiveness of the MCP Authorization Model for LLM-powered agents.

II. Related Work

In [9], the authors introduced the universal agent protocol, which became the foundation for modern agent systems, including LangChain Agents. In [10], the authors presented plugin-based interfaces such as OpenAI ChatGPT Plugins, enabling AI models to connect with external tools through standardized API schemas like OpenAPI. In [11], the authors explored contextual information retrieval methods, such as retrieval augmented generation (RAG) and knowledge bases, enabling models to supplement responses with up-to-date information by leveraging vector-based search to retrieve relevant knowledge from databases. In [12], the authors defined standardized interface through JSON-RPC message exchange and the modal context protocol (MCP), extending passive information retrieval by enabling AI models to interact with external data sources and addressing the fragmentation problem in AI tool integration. Despite the enhanced capabilities of LLMs and recent advancement in AI agents interactions tools, existing research fails to demonstrate the effectiveness of MCP for AI agent interaction within a zero-trust framework.

III. MCP Background and Security Challenge

The Model Context Protocol (MCP) is an open standard that enables secure and uniform access to external tools and services for large language models [13]. It acts as a mediator between the LLM and external tools. MCP interaction includes roles such as MCP clients, which are AI agent or LLMs, and MCP server, which represent external tool or data sources. Real-time MCP use cases include software development assistant and AI-driven support assistants. MCP enables enterprise to integrate AI with existing applications.

However, security challenges could arise if MCP is not designed correctly. Common MCP security challenges include overprivileged access, supply chain exposure, inconsistent policy enforcement, and context leakage [14]. To secure MCP, industry standard recommendations would benefit organizations deploying it. These include applying fine-grained access control, introducing AI governance into AI workflows, avoiding static credentials, enforcing mandatory logging and audit monitoring, and educating both developer and security teams.

IV. Methodology

The study leverages machine learning approach to assess the MCP authorization model effectiveness. The experimental methodology presented in this study incorporates AI-driven Azure services logs to construct a sample dataset that includes features such as source_ip, destination_ip, event_type, alert_severity, and target_variable. These dataset features are chosen to identify both network-based and securty-relevant identifiers. The initial preprocessing involves cleaning to remove duplicate entries, label encoding to convert categorical values into numerical values, and normalization (StandardScaler) to scale numerical values to ensure uniformity across features, followed by exploratory data analysis (EDA) to identify patterns and security indicators. The full dataset is then divided into training (80%) and testing (20%) segments, with Synthetic Minority Oversampling Technique (SMOTE) applied to address class imbalances. Machine learning model are selected, trained, tuned, and deployed to assess effectiveness of the MCP authorization model for LLM-powered agent. Model performance is evaluated using appropriate metrics.

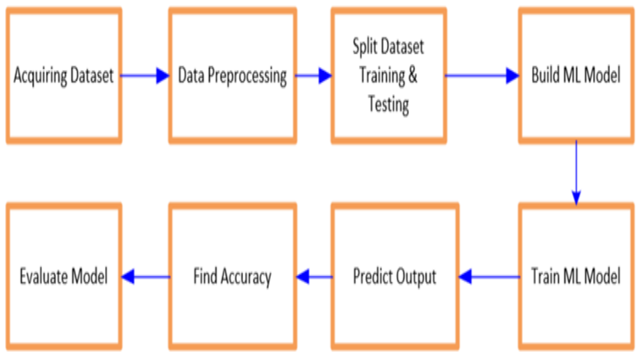

Fig. 1. Machine learning implementation workflow.

Fig. 1 shows the visual workflow of machine learning implementation process, which involves several key stages: acquiring the dataset, followed by data preprocessing, splitting the dataset into training and testing sets, building and training the model, predicting outputs, evaluating accuracy, and finally assessing the model’s overall performance.

A modular ML pipeline was designed with the following components:

- Feature Extraction: Using AI-driven Azure service logs.
- Classification Methods and Model Architecture: RM, KNN, SVM, Gradient Boosting, and Logistic regression.
- Model Evaluation: Assessing accuracy of Machine learning models to evaluate their performance.

V. Responsible AI Consideration

Although the dataset used in this research is synthetic and does not contain sensitive information, ensuring responsible AI practices is important. Future work will focus on integrating explainable AI (XAI) tools such as SHAP to enhance transparency and trust in ML-driven authorization decisions, which will help align the models with ethical standards and enterprise governance requirements.

VI. Results and Discussion

Proposed study utilizes exploratory data analysis (EDA) techniques to understand the dataset and detect any anomalies prior to training a machine learning model. Fig. 2 shows the correlation heatmap that visualizes correlation coefficient between source_ip, destination_ip, event_type, alert_severity, and target_variable. These variables show weak correlations with each other and with target variable, indicating that liner models may not perform well. Although the heatmap highlights that the correlation between event type and alert_severity is the highest at 0.12 and may offer slightly more predictive value, it still considers a weak correlation. Given the overall weak correlations, a Random Forest model is considered a good choice since it is robust to noise and handles weakly correlation features effectively.

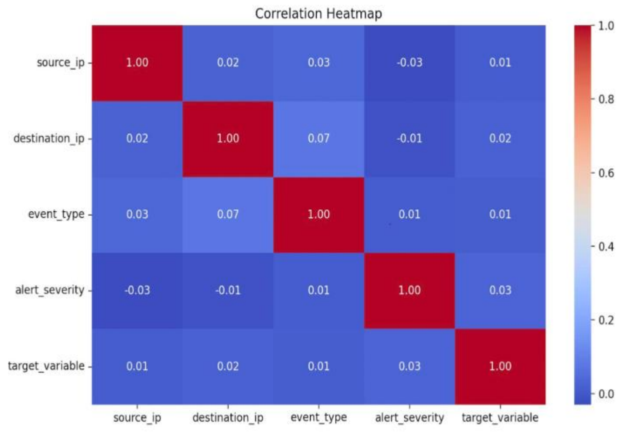

Fig. 2. Correlation heatmap.

Fig. 3 shows box plots illustrating the feature distributions of source_ip, destination_ip, event_type, alert_severity, and target_variable. Most of these features are tightly centered around 0, suggesting that they are well preprocessed for machine learning.

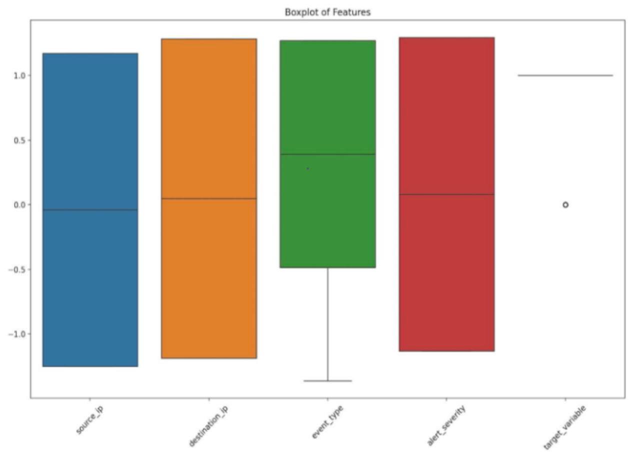

Fig. 3. Box plot for feature distribution.

Fig. 4 shows six numerically encoded features, where source_ip and destination_ip indicate various network traffic patterns, event_type and alert_severity show imbalanced frequencies, and target_variable, which is heavily skewed toward the malicious class indicates class imbalance.

Fig. 5 displays class distribution before and after applying SMOTE (Synthetic Minority Over-Sampling Technique). Class 0 has fewer samples (100) compare to class 1 (600), which causes models to favor the majority class, leading to biased predictions. To address class imbalance issues, SMOTE has been applied, enabling machine learning models avoid bias toward the majority class and improve overall performance.

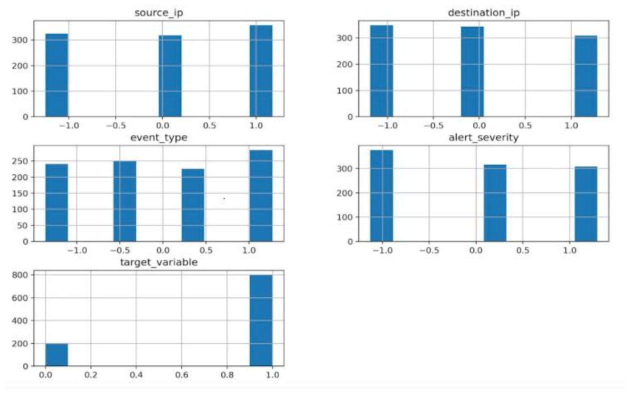

Fig. 4. Bar charts showing categorical feature distributions.

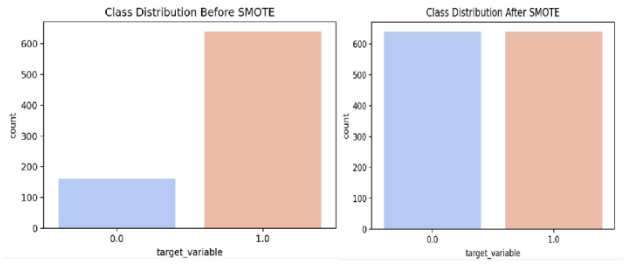

Fig. 5. Class distribution before and after SMOTE.

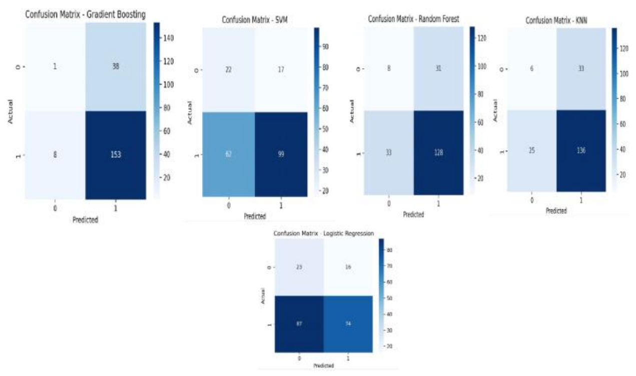

Fig. 6. Consolidated confusion matrix.

Fig. 6 shows the consolidated confusion matrix for five models. Based on the sample outlined in the confusion matrix, machine learning models produced the following results:

$$Accuracy\ (A) = \frac{tp+tn}{tp+tn+fp+fn} \quad (1)$$

$$Precision\ (P) = \frac{tp}{tp+fp} \quad (2)$$

$$Recall\ (R) = \frac{tp}{tp+fn} \quad (3)$$

$$F1 - Score = \frac{2*(p*r)}{p+r} \quad (4)$$

where, TP is true positive, TN is true negative, FP is false positive, and FN is false negative.

TABLE I. MODEL METRICS

Random forest	SVM	KNN	Gradient boosting	Logistic regression

TP= 128
TN= 8
FP= 31
FN= 33
A= 0.68
P= 0.805
R= 0.795
F1-Score= 0.800
	TP= 99
	TN=22
	FP= 17
	FN=62
	A=0.6 P= 0.853
	R= 0.615
	F1-Score= 0.714
		TP= 136
		TN= 6
		FP= 33
		FN= 25
		A= 0.71
		P= 0.805
		R= 0.845
		F1-Score= 0.824
			TP= 153
			TN= 1
			FP= 38
			FN= 8
			A= 0.77
			P= 0.801
			R= 0.950
			F1-Score= 0.869
				TP= 74
				TN= 23
				FP= 16
				FN= 87
				A= 0.485
				P= 0.822
				R= 0.459
				F1-Score= 0.589

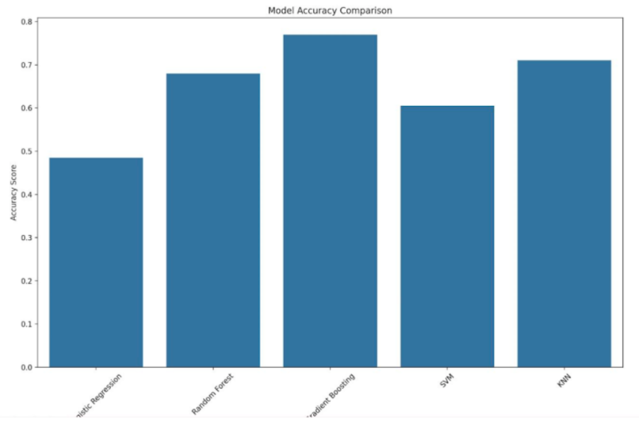

Fig. 7. Model accuracy comparison.

Five algorithms are considered on this dataset to perform classification tasks. The model is evaluated using key parameters such as Recall, Precision, F-Measure, error rate and overall model accuracy. Gradient Boosting was chosen for assessing the effectiveness of the MCP Authorization model for LLM-powered agents for the following reasons:

- Gradient Boosting achieved the highest score among all other models tested in this study.
- Gradient Boosting handles complex feature interaction effectively and its sequential assemble approach, which corrects errors from previous iterations, is beneficial for identifying malicious patterns in the MCP server log.

Table I details the model metrics and the accuracy comparison of models is given in Fig. 7.

VII. CONCLUSION

This study presents a machine learning-driven evaluation of the Model Context Protocol (MCP) authorization model to enhance secure communication for LLM-powered agents. The experimental methodology used in this research study leverages an ML pipeline to evaluate the MCP authorization model using AI-driven Azure service logs, including Azure Monitor, Azure Sentinel, Azure Active Directory, which are used to monitor MCP server activity. These logs are used to construct a dataset, containing 5,000 labeled instances of observed attacks outcomes across seven features, including source_ip, destination_ip, event_type, alert_severity, and target_variable, to train a model to identify anomalous activities in AI-agent communication with model. With an accuracy of 77 per cent, this study highlights the effectiveness of ML technique in predicting the performance of the MCP authorization model for LLM-powered agent, enabling organizations to better understand the importance of secure connection between AI models and enhance the security of AI-powered applications.

VIII. Limitations and Future Work

Although the proposed ML pipeline shows promise, the research can be enhanced by applying larger dataset. Future work should focus on incorporating additional data source to improve model robustness, integrating explainable AI (XAI) tools such as SHAP to enhance transparency and trust in ML-driven authorization decisions, extending the model to other cloud platforms beyond Azure, evaluating the MCP framework to assess its resilience, and deploying the ML pipeline in live enterprise settings under operational constraints.

References

[1] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, et al., "GPT-4 Technical Report," arXiv preprint arXiv:2303.08774, 2023.

[2] A. Grattafiori, A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Vaughan, et al., "The LLaMA 3 Herd of Models," arXiv preprint arXiv:2407.21783, 2024.

[3] A. Yang, A. Li, B. Yang, B. Zhang, B. Hui, B. Zheng, B. Yu, C. Gao, C. Huang, C. Lv, et al., "Qwen3 Technical Report," arXiv preprint arXiv:2505.09388, 2025.

[4] LangChain, "LangChain: Framework for Developing Applications Powered by Language Models," 2022. Available: https://github.com/langchain-ai/langchain

[5] J. Liu, "LlamaIndex: A Data Framework for LLM Applications," 2022. Available: https://github.com/run-llama/llama_index

[6] Anthropic, "Introducing the Model Context Protocol," 2024. Available: https://www.anthropic.com/news/model-context-protocol

[7] N. Gunasinghe and N. Marcus, Language Server Protocol and Implementation. Springer, 2021.

[8] Anthropic, "Introducing the Model Context Protocol," Blog post, November 2024. Available: https://www.anthropic.com/news/model-context-protocol

[9] H. Chase, "LangChain," Python framework for developing applications powered by language models, 2022. Available: https://github.com/langchain-ai/langchain

[10] OpenAI. 2023. ChatGPT plugins. https://openai.com/index/chatgpt-plugins/

[11] F. Cuconasu, G. Trappolini, F. Siciliano, S. Filice, C. Campagnano, Y. Maarek, N. Tonellotto, and F. Silvestri, "The Power of Noise: Redefining Retrieval for RAG Systems," CoRR, abs/2401.14887, 2024. doi: 10.48550/arXiv.2401.14887

[12] T. Gan and Q. Sun, "RAG-MCP: Mitigating Prompt Bloat in LLM Tool Selection via Retrieval-Augmented Generation," arXiv preprint arXiv:2505.03275, 2025.

[13] Anthropic, "Model Context Protocol," 2024. Available: https://docs.anthropic.com/en/docs/agents-and-tools/mcp. Accessed: June 25, 2025.

[14] S. Walter, Securing the Model Context Protocol: Access, Authorization, and Audit for Enterprise AI. Hyperframe Research, 2025. Available: https://hyperframeresearch.com/2025/05/21/securing-the-model-context-protocol-access-authorization-and-audit-for-enterprise-ai/.