ESANN 2026 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 22-24 April 2026, i6doc.com publ., ISBN 9782875870964. Available from http://www.i6doc.com/en/.

Model Selection Hijacking Adversarial Attack

Luca Pajola[1], Riccardo Petrucci[2], Francesco Marchiori[3], Luca Pasa[2,3], Mauro Conti[1,2,3,4]

1- SpritzMatter Srl, Padova, Italy

2- Department of Mathematics - University of Padova, Padova, Italy

3- Human Inspired Technology Centre (HiT) - University of Padova, Italy

4- Örebro University, Örebro, Sweden

Abstract. Model selection plays a critical role in the deployment of machine learning systems, yet its vulnerability to adversarial manipulation remains largely unexplored. We introduce **MOSHI** (**MO**del **S**election **HI**jacking), a novel framework that examines whether targeted poisoning of only the validation set, without any access to training data, model internals, or system configuration, can systematically bias the selection process toward inferior models. Leveraging a VAE-based perturbation mechanism, we empirically demonstrate that MOSHI can induce coherent misselection in both vision and speech benchmarks, leading to models with degraded generalization, as well as increased inference latency and energy consumption. Our results highlight that model selection, typically viewed as a benign step, can significantly affect robustness, suggesting it should be treated as an integral component of adversarial ML analysis.

1 Introduction

As machine learning systems transition from idealized theoretical settings to practical deployment, growing emphasis has been placed on understanding how real-world constraints and adversarial conditions influence their reliability and fundamental properties. Within this landscape, Adversarial Machine Learning (AML) has exposed how ML pipelines can be manipulated through evasion during inference [1], data poisoning during training [2], or privacy-oriented attacks such as membership inference [3]. However, AML research has focused almost exclusively on the training and inference phases. Much less attention has been given to the intermediate steps that drive deployment decisions-particularly *model selection*. Model selection, in principle, aims to identify hyperparameters that promote strong generalization, but the chosen hyperparameters also influence model efficiency, computational cost, and other key behavioral properties that affect downstream application and usage. With this perspective, it becomes evident that the model selection process itself constitutes a potential attack surface, susceptible to exploitation in ways that can degrade model quality or impose harmful computational and operational consequences on the deploying platform. Many practical ML workflows rely on fixed dataset splits, especially in open-source platforms like Kaggle and Hugging Face, where training and validation sets are reused across hundreds of projects. Poisoning these splits, even subtly, could bias model selection across many downstream applications. In Machine-Learning-as-a-Service (MLaaS) scenarios, validation data is often client-provided, making it an attractive and realistic target. A malicious provider could manipulate it to promote models that

ESANN 2026 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 22-24 April 2026, i6doc.com publ., ISBN 9782875870964. Available from http://www.i6doc.com/en/.

are more expensive to run (e.g., in energy or latency), thereby increasing operational costs or nudging deployments toward architectures that leak more information.

In this work, we investigate how the model selection process can be poisoned to negatively affect various aspects of model behavior. Despite its critical influence on both performance and operational characteristics, model selection poisoning has received little attention in the existing literature under adversarial settings. Unlike data poisoning attacks, an adversary does not need to modify training data or interfere with the training process. Instead, influencing only the validation set could silently bias the model chosen for deployment, favoring models that are slower, less accurate, or more resource-hungry. To demonstrate the feasibility and potential risks of model selection poisoning, we introduce a novel framework called Model Selection Hijacking **(MOSHI)**. MOSHI employs a Variational Autoencoder-based approach to generate adversarial validation samples that systematically bias automated model ranking, steering the selection process toward suboptimal models, such as those with slower inference or reduced generalization, without modifying training data or model internals. We evaluated the proposed approach on both computer vision and speech recognition tasks under various conditions and experimental settings. The empirical findings confirm its effectiveness, unveiling a previously overlooked vulnerability in the machine learning deployment pipeline.

2 Background and Related Works

In this section, we review the adversarial ML attacks from recent years that are related to the MOSHI framework presented in this work. These include various approaches aimed at influencing model behavior during training or evaluation.In Adversarial Machine Learning, model poisoning manipulates the training process by injecting maliciously crafted data into the training set, corrupting the learning algorithm and leading to compromised performance [4]. Such attacks can subtly alter decision boundaries, degrade accuracy, or embed vulnerabilities, allowing attackers to influence future predictions. By exploiting trust in training data, model poisoning underscores the need for securing and validating ML workflows. During the deployment phase, ML models, especially Deep Neural Networks (DNNs), require significant computational resources and memory. To address these demands, specialized hardware such as Big Basin [5], Project BrainWave [6], and Tensor Processing Units (TPUs) [7] are developed, referred to as Application-Specific Integrated Circuits (ASIC). Shumailov et al. [8] initiated AML research focused on increasing resource consumption (e.g., energy, latency) of target models, introducing the *sponge examples attack*. This attack generates adversarial "sponge" samples that escalate latency and energy consumption, aiming to induce an availability breakdown rather than affecting prediction accuracy. The attack works by manipulating *(i)* the computation dimension, increasing time complexity, or *(ii)* data activation sparsity, causing more operations in ASICs by reducing activation sparsity, with applications in Natural Language Processing and Computer Vision. Cina et al. [9] proposed a poisoning attack designed to increase latency and energy consumption without affecting prediction performance. This attack uses sponge training, a model poisoning technique, to optimize energy consumption

ESANN 2026 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 22-24 April 2026, i6doc.com publ., ISBN 9782875870964. Available from http://www.i6doc.com/en/.

by increasing model activation, thus undermining ASIC accelerators' efficiency.

MOSHI and poisoning attacks. Unlike traditional poisoning attacks, MOSHI never interacts with the training dataset nor changes model parameters. Instead, it biases validation loss computation so that models scoring better according to an attacker-defined *hijack metric* are artificially favoured.

3 Model Selection Hijacking Attack

We focus our analysis on the common supervised learning setting, where the training phase aims to learn a function that maps inputs $\boldsymbol{x} \in \mathcal{X}$ to labels $y \in \mathcal{Y}$ based on a finite labeled training set $\mathcal{S} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n}$ sampled from an unknown distribution $\mathcal{D}$. A learning algorithm $\mathscr{A}$ selects a hypothesis $h \in \mathcal{H}$ that approximates the unknown labeling function and produces predictions $\hat{y} = h(\boldsymbol{x})$.Model selection focuses on identifying the best hypothesis h among multiple candidates by tuning hyperparameters that define $\mathscr{A}$'s behavior. Each candidate $\mathscr{A}_{\mathfrak{c}}$ has different hyperparameter configurations $\mathfrak{c} \in \mathfrak{C}$, and the resulting models are ranked according to their estimated error on a validation set $\mathcal{S}^{Val}$. Because evaluating many configurations is costly, grid search and other structured strategies are often used to limit the search space. When data is limited, resampling strategies such as the Hold-Out method or k-fold cross-validation are employed to maximize data efficiency. In this work, we focus on the Hold-Out approach, where all models are trained on the same subset $\mathcal{S}^{Train}$ and the best model is selected as the minimum validation loss on $\mathcal{S}^{Val}$.

MOSHI compromises the model selection phase by replacing part of the validation set $\mathcal{S}^{Val}$ with a set of adversarial samples $\mathcal{S}^{Val}_{pois}$ generated by the attacker. The selection procedure then incorrectly chooses the model $\tilde{h}_{\mathfrak{c}^*}$ that performs best on the poisoned validation data: $\tilde{h}_{\mathfrak{c}^*} = \text{argmin}_{\mathfrak{c} \in \mathfrak{C}} \mathcal{L}_{Val}(h_{\mathfrak{c}}, \mathcal{S}^{Val}_{pois})$.

3.1 Adversarial Sample Generation

To generate $\mathcal{S}^{Val}_{pois}$, we introduce a modified Conditional Variational Autoencoder (HVAE). Standard VAEs map samples to a latent space through an encoder and reconstruct them via a decoder, optimized using reconstruction loss $\mathcal{L}_{rec}$ and KL divergence $\mathcal{L}_{KLD}$. HVAE extends this formulation by incorporating the hijack objective directly into its loss function $\mathcal{L}_{HVAE} = \left(\mathcal{L}_{rec} + \mathcal{L}_{KLD} - Hj_{cost}(\mathfrak{C})\right)^2$, where $Hj_{cost}(\mathfrak{C}) = \frac{1}{|\mathfrak{C}|} \sum_{\mathfrak{c} \in \mathfrak{C}} \alpha \cdot \mathcal{L}_{Val}(h_{\mathfrak{c}}, \mathcal{S}_{gen})$, and

$$\alpha = \frac{\max_{\lambda \in \Lambda} \{m(h_\lambda, \mathcal{S}^{Val})\} - m(h_{\mathfrak{c}}, \mathcal{S}^{Val})}{\max_{\lambda \in \Lambda} \{m(h_\lambda, \mathcal{S}^{Val})\} - \min_{\lambda \in \Lambda} \{m(h_\lambda, \mathcal{S}^{Val})\}}.$$

The new term $Hj_{cost}(\mathfrak{C})$ pushes the HVAE to produce samples that increase validation loss for models with *low* hijack metric values while minimally affecting models with *high* metric values. As a result, the generated samples systematically distort validation ranking, steering model selection toward the attacker's desired configuration. Figure 1 provides a graphical overview of the attack pipeline and the generation of $\mathcal{S}^{Val}_{pois}$.

ESANN 2026 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 22-24 April 2026, i6doc.com publ., ISBN 9782875870964. Available from http://www.i6doc.com/en/.

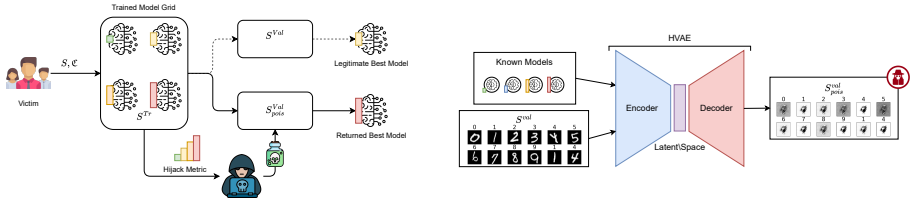

Fig. 1: Schematic representation of the overall pipeline of the proposed attack and the generation process of $\mathcal{S}^{Val}_{pois}$.

3.2 Hijack Metric

MOSHI allows different attack objectives by defining a hijack metric m that determines which model should appear "best". We explore three classes of metrics: *(i)* **Generalization Attack:** selects a model that performs poorly on unseen data (underfitting/overfitting); *(ii)* **Latency Attack:** selects models with higher computational latency; *(it)* **Energy Consumption Attack:** selects models that maximize computational or activation cost. Each of these metrics requires defining a corresponding $Hj_{cost}(\mathfrak{C})$ term tailored to the specific attack objective. For generalization, we use the loss on the original validation set. Latency is measured as the total processing time for a given dataset. Energy consumption is approximated using an ASIC simulator [8, 9], and when incompatible with the SpeechCommands models, we use the ℓ_0 norm of neuron activations, computed as the mean number of non-zero activations across all layers.

4 Experimental Evaluation

MOSHI is evaluated through three classification case studies spanning computer vision and speech recognition: MNIST and CIFAR10 for vision, and Speech Commands for speech [10]. The victim models vary across tasks, including FeedForward Neural Networks, DenseNets [11], and CNNs, with different architectural configurations such as depth and final layer size.

We explore the attack effect in two fashions: *(i) Full substitution*, where the validation set is entirely substituted by the malicious samples; *(i) Partial substitution*, where the validation set combines both legitimate and malicious samples at different rates.

To evaluate the strengths of the proposed approach, we consider two threat settings. In the *White-Box* (WB) scenario, the attacker has full access to all candidate models, including their architectures and parameters In contrast, the *Black-Box* (BB) scenario assumes that the attacker only knows the architecture search space, while the internal details of the models remain hidden. To facilitate reproducible experimentation and encourage deeper exploration of model selection within ML pipelines, we make our code publicly available[1]. For full substitution, table 1 shows shows the ratio between metrics from standard model selection and those from a hijacked selection using HVAE, *e.g.* in the white-box MNIST case, the selected model was up to 3× slower

[1]https://github.com/pajola/MSHAA

ESANN 2026 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 22-24 April 2026, i6doc.com publ., ISBN 9782875870964. Available from http://www.i6doc.com/en/.

in latency. White-box and black-box settings yield similar effects on MNIST and SpeechCommands, with the white-box showing better generalization. For CIFAR10, the black-box scenario slightly outperforms the white-box scenario in terms of latency impact. From the results, it is possible to observe that, in many cases, poisoning the validation set negatively impacts model behavior according to the considered hijacking metrics. However, our experiments show a negligible impact on model classification performance (except for the case of the generalization metric).

Setting	Metric	MNIST	CIFAR10	SpeechC.
White-Box	Gen.	21.27	1.12	4.14
	Lat.	3.82	0.77	0.80
	Eng.	1.24	1.01	N/A
	ℓ_0	6.29	1.03	0.63
Black-Box	Gen.	21.27	1.18	2.32
	Lat.	3.82	1.00	0.80
	Eng.	1.24	1.01	N/A
	ℓ_0	6.29	1.03	0.81

Table 1: Ratio between metrics from standard and hijacked model selection. A value of 1 means no impact, while a value >1 indicates degraded performance.

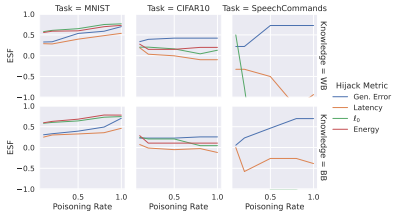

Fig. 2: White-Box (WB) and Black-Box (BB) poisoning rate.

For Partial substitution, Figure 2 reports the three use cases, the four hijack metrics, and both knowledge levels. To qualitatively assess the attack's effectiveness on the selected model grid, we introduce two novel metrics: the *Effectiveness Score Function* (ESF). ESF measures the normalized difference between a model metric and the performance obtained by selecting the model using the original validation set:

$$ESF(\mathfrak{C}) = \frac{\mathcal{E}(\tilde{h}_{\mathbf{c}^*}, \mathcal{S}^{Test}) - \mathcal{E}(h_{\mathbf{c}^*}, \mathcal{S}^{Test})}{\max_{\mathbf{c} \in \mathfrak{C}} \{\mathcal{E}(h_{\mathbf{c}}, \mathcal{S}^{Test})\} - \mathcal{E}(h_{\mathbf{c}^*}, \mathcal{S}^{Test})},$$

where $\mathcal{E}$ is the chosen metric, $h_{\mathbf{c}^*}$ is the model using $\mathcal{S}^{Val}$ and $\tilde{h}_{\mathbf{c}^*}$ using $\mathcal{S}^{Val}_{pois}$. We can observe that the white-box settings help produce a more potent attack comparable to more traditional poisoning properties. Despite that, we do not observe clear advantages in poisoning the validation set entirely. This result suggests that HVAE can effectively attack even by tampering with only a smaller portion of the validation set. On the other hand, in the Speech Commands case study, we observe a more erratic behavior of the mean ESF. This is particularly evident for the ℓ_0 hijack metric.

5 Conclusion

Model selection is a critical yet often overlooked phase in the machine learning pipeline, central to achieving optimal performance, robustness, and scalability. In this work, we introduce a novel perspective: that the validation set itself can become an attack surface. We present MOSHI, to our knowledge the first adversarial technique that targets model selection by poisoning the validation set. MOSHI demonstrates the **feasibility** of such an attack through a generative approach based on the Hijacking VAE (HVAE). This method injects carefully crafted validation examples to steer model selection toward a less optimal outcome, either in terms of performance or computational overhead, without modifying the training procedure or loss function.

ESANN 2026 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 22-24 April 2026, i6doc.com publ., ISBN 9782875870964. Available from http://www.i6doc.com/en/.

As future work, we plan to extend the evaluation by considering additional validation methods beyond the hold-out procedure, such as cross-validation.

References

[1] Battista Biggio, Igino Corona, Davide Maiorca, Blaine Nelson, Nedim Šrndić, Pavel Laskov, Giorgio Giacinto, and Fabio Roli. Evasion attacks against machine learning at test time. In *Machine Learning and Knowledge Discovery in Databases*. Springer Berlin Heidelberg, 2013.

[2] Battista Biggio, Blaine Nelson, and Pavel Laskov. Poisoning attacks against support vector machines. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, ICML'12, 2012.

[3] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE symposium on security and privacy (SP)*. IEEE, 2017.

[4] Zhiyi Tian, Lei Cui, Jie Liang, and Shui Yu. A comprehensive survey on poisoning attacks and countermeasures in machine learning. *ACM Computing Surveys*, 55(8), 2022.

[5] Kim Hazelwood, Sarah Bird, David Brooks, Soumith Chintala, Diril, et al. Applied machine learning at facebook: A datacenter infrastructure perspective. In *2018 IEEE international symposium on high performance computer architecture*. IEEE, 2018.

[6] Eric Chung, Jeremy Fowers, Kalin Ovtcharov, Michael Papamichael, Adrian Caulfield, Todd Massengill, Ming Liu, Daniel Lo, Shlomi Alkalay, Michael Haselman, et al. Serving dnns in real time at datacenter scale with project brainwave. *iEEE Micro*, 38(2), 2018.

[7] Norman Jouppi, Cliff Young, Nishant Patil, and David Patterson. Motivation for and evaluation of the first tensor processing unit. *ieee Micro*, 38(3):10–19, 2018.

[8] Ilia Shumailov, Yiren Zhao, Daniel Bates, Nicolas Papernot, Robert Mullins, and Ross Anderson. Sponge examples: Energy-latency attacks on neural networks. In *2021 IEEE European symposium on security and privacy (EuroS&P)*, pages 212–231. IEEE, 2021.

[9] Antonio Emanuele Cinà, Ambra Demontis, Battista Biggio, Fabio Roli, and Marcello Pelillo. Energy-latency attacks via sponge poisoning. *arXiv preprint arXiv:2203.08147*, 2022.

[10] Pete Warden. Speech commands: A dataset for limited-vocabulary speech recognition, 2018.

[11] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.