NERVE Attacks: Breaking AI-Powered Brain-Computer Interfaces

Zahra Tarkhani*‖, Georgios Akkogiounoglou†‖, Lorena Qendro‡, Isabel Tscherniak§‖, Anil Madhavapeddy¶
*Microsoft
†KTH Royal Institute of Technology
‡Nokia Bell Labs
§Technical University of Munich
¶University of Cambridge

***Abstract*—The rapid integration of AI into human-centred systems such as Brain-Computer Interfaces (BCIs) has created a poorly understood attack surface linking neural signals to physical systems. Exploits in this domain threaten cognitive autonomy, mental privacy, and physical safety—from neural data exfiltration to malicious control of BCI-tethered devices. We introduce the NERVE Attacks class, a systematic characterisation of five orthogonal attack dimensions that together span the complete BCI stack: Neuro-mimetic Forgery (N), Evasion via Desynchronization (E), Replay-based Hijacking (R), Vein Tapping (V), and Embedded Backdoors (E). To evaluate this class we present EEGle, an AI-assisted extensible framework for systematic BCI security analysis. Our evaluation uncovers 17 novel neuro-specific attack instances and reveals a stealth-effectiveness spectrum unique to BCI backdoor design. We also show that generative AI lowers the barrier to entry for non-expert attackers, and provide EEGle to the community for building and verifying the security of these deeply personal devices.**

I. Introduction

The convergence of neuroscience, artificial intelligence (AI), microelectronics, and wearable systems is fueling a new generation of wearable human-computer interaction (HCI) devices with Brain-Computer Interfaces (BCIs) at the forefront. BCI-assisted applications are now extending beyond traditional healthcare and medical uses, such as prosthetic control and mental health [1]–[5], into seamlessly AI-integrated computing in wearables, autonomous vehicles, robotics, thought-based communication, and augmented reality use cases such as Synchron's BCI integration with Apple's Vision Pro [6]–[11].

However, this rapid adoption has also revealed a novel and poorly understood attack surface [12]–[14]. The highly personal nature of neural data, coupled with the potential for direct control over brain-controlled prosthetics and systems, significantly raises the stakes of cybersecurity beyond those of conventional HCI and wearable technologies. An exploit in this domain is not a mere data breach or system crash; it directly threatens a user's cognitive autonomy, mental privacy, and physical safety [12], [15]–[18].

BCI systems operate on continuous physiological time-series data that are noisy, low signal-to-noise ratio, non-stationary, and highly subject-specific, which makes both model behaviour and adversarial effects more difficult to interpret, constrain, and validate than in image- or text-based models [19]–[21]. At the same time, BCIs do not exist in isolation. They sit within a broader ecosystem of headsets, operating systems, cloud/mobile services, and actuated devices. Vulnerabilities at the BCI layer can propagate across this stack and can bypass or undermine security controls that were never designed for neural data paths or closed-loop actuation. As a result, current AI security models and defences are inadequate for this emerging neuro-specific threat landscape [11], [12].

This work is driven by a twofold critical hypothesis concerning the inherent vulnerabilities of these new physiological computing systems. First, we hypothesise that BCI systems are uniquely vulnerable to new classes of domain-specific semantic attacks and particularly adversarial evasion and backdoors that are crafted to target the integrity of human intent. Attacks on BCI models must be physiologically plausible; an attacker cannot simply add arbitrary noise without detection. Malicious input must materialise as either a subtle, replayable neural signal or as a weaponised external stimulus that evokes a predictable brain response. We posit that attackers will exploit the unique *neuro-specific* properties of BCI data (e.g., frequency-based artefacts, event-related potentials) to craft evasion and backdoor attacks that are both highly effective and stealthy.

Second, we hypothesise that the barrier to entry for creating these sophisticated, neuro-specific attacks is collapsing rapidly. Historically, developing such exploits would require rare cross-domain expertise in neuroscience, signal processing, and AI security. However, the rise of powerful generative AI, including Large Language Models (LLMs) and diffusion models, is accelerating this threat. We argue that a non-expert attacker can now leverage generative tools to create physiologically plausible attack payloads, craft malicious training data, or write exploit code, all without deep BCI knowledge.

We argue that both hypotheses are *compounded* by a persistent failure to address foundational system-level insecurities. Unencrypted communication, missing authentication, inadequate access control, and absent defence-in-depth mechanisms

‖The majority of this work was conducted while these authors were Visiting Researchers at the University of Cambridge.

remain widespread and now serve as the delivery vector for novel ML-based exploits—a cross-layer threat that has not yet been holistically evaluated.

This combination of a high-stakes attack surface and a collapsing barrier to entry represents an imminent and severe threat. To close the gap, we introduce the *NERVE Attacks*: five orthogonal BCI-specific attack dimensions that together span the critical components of modern AI-powered BCIs, providing a novel and structured view of the threat landscape. (Section IV). To systematically evaluate our hypothesis and prototype these attack vectors, we implemented *EEGle*, the first extensible framework for security analysis of AI-powered BCIs. In summary, we make the following contributions:

1) **NERVE Attacks.** We define five orthogonal dimensions spanning the complete BCI attack surface: **N**euromimetic Forgery, **E**vasion via Desynchronization, **R**eplay-based Hijacking, **V**ein Tapping, and **E**mbedded Backdoors. Within these dimensions we discover and formalise 17 novel neuro-specific attack instances (summarized in Table I), including entirely new attack families (NFA, TDA, NRA) and a ten-trigger BCI backdoor suite, each exploiting physiological properties absent from standard adversarial ML toolboxes and prior BCI security work.
2) **Neuro-specific evasion attacks.** We introduce Neuromimetic Forgery Attacks (NFA) and Neuro Replay Attacks (NRA), showing that generative AI lowers the barrier to entry, enabling non-expert attackers to achieve targeted, low-visibility manipulation of model outputs. We also introduce Temporal Desynchronization Attacks (TDA), demonstrating architecture-dependent brittle failure under low-cost timing shifts.
3) **BCI backdoor suite with stealth-effectiveness spectrum.** We introduce a suite of BCI-specific model backdoors and a systematic backdoor-injection methodology. Our analysis reveals a fundamental stealth-effectiveness trade-off unique to BCI backdoor design, providing an attacker menu ranging from high-stealth/high-ASR implants to lower-stealth backdoors with 100% attack success rate (ASR).
4) **EEGle framework.** We present EEGle, an innovative and extensible security analysis framework for investigating the complex and evolving threat landscape of wearable AI-integrated BCI applications, and provide it to the community as a foundational tool for building and verifying the security of these deeply personal devices.

II. Background and Related Work

A. AI-Powered BCI Pipeline

Contemporary non-invasive BCI systems transform neural signals into commands through a multi-stage pipeline. At the physical layer, an EEG headset captures cortical potentials (20–100 μV) with excellent temporal resolution ($\sim$1 ms) but poor spatial resolution due to volume conduction. Systems decode user intent via endogenous paradigms—Motor Imagery (MI), which modulates mu/beta rhythms through Event Related Desynchronisation (ERD) and Synchronisation (ERS) [22], [23]—and exogenous paradigms such as steady-state visual evoked potential (SSVEP) and P300 event-related potential (ERP). Preprocessing enhances the signal-to-noise ratio via temporal and spatial filtering (e.g., Common Average Referencing) and artifact rejection (e.g., ICA) [24], [25]; Common Spatial Patterns (CSP) then extract discriminative features for MI [26].

Classification has shifted from interpretable CSP + linear discriminant analysis (LDA) or support vector machine (SVM) pipelines [27] to end-to-end deep learning (EEGNet [28], DeepSleepNet [29]) and, most recently, to self-supervised brain foundation models trained on large unlabeled EEG corpora (BIOT [30], NeuroLM [31], LaBraM [32], EEGPT [33], CBraMod [34]). These larger, increasingly opaque AI pipelines expand the attack surface of deeply personal devices and motivate a systematic security analysis.

B. Security Gaps and Prior Work

BCI security. Consumer BCI systems have long been known to lack various fundamental security protections—cryptographic hardening, privilege separation, and authenticated device pairing—leaving both the wireless link and host software stack independently exploitable [12], [14], [35]–[38]. Passive observation of EEG streams can leak sensitive cognitive content [39], and commercial devices have been shown to enable subliminal probing of private mental state [40]. Unencrypted channels and absent device authentication are pervasive across BCI hardware [41]–[44], while memory-unsafe SDK implementations [45] and the complete absence of supply-chain provenance tooling [46]–[48] compound the risk at the host level.

Yet none of this body of work addresses the AI-specific attack surface introduced by modern AI pipelines: the five orthogonal dimensions of NERVE Attacks—physiologically plausible signal forgery, timing-based evasion, wireless replay hijacking, passive eavesdropping, and persistent model backdooring—remain entirely uncharacterised as a unified threat class prior to this work.

Adversarial and backdoor attacks on EEG.

Prior adversarial attacks on EEG classifiers routinely fail to survive modern BCI preprocessing [15]–[17], [49], and existing backdoor work remains confined to isolated model-level analyses [50]–[53]. Our evasion and backdoor attacks introduce novel threat families absent from the broader adversarial ML literature, and no prior work provides a framework for systematically mounting and extending them; EEGle fills this gap.

Semantic plausibility and GenAI. Crafting physiologically plausible attack payloads historically required rare multi-domain expertise in neuroscience, signal processing, and AI security, an implicit barrier now collapsing. Powerful generative models enable non-expert attackers to produce realistic EEG signals and synthesise novel backdoor triggers [20], [21]. Our work is the first to utilize multimodal LLMs like Claude or Gemini as a *constrained synthetic data and parameter genera-*

TABLE I: Summary of the 17 novel attack instances comprising NERVE Attacks, detected and explored using EEGle (details are explained in Section VII). **Dim.** is the NERVE dimension the instance belongs to: N = Neuro-mimetic Forgery, E = Evasion via Desynchronization, R = Replay-based Hijacking, V = Vein Tapping, and Ebd = Embedded Backdoors. **Layer** is the level of the BCI stack the instance targets (Figure 1): *Transport* (BLE/Wi-Fi link), *Host* (SDK, sockets, filesystem), *Input* (the signal as presented to preprocessing), or *Model* (the trained classifier itself).

ID	Dim.	Layer	Attack	Description
A1	N	Input	NFA-I	Synthetic brain signal fools BCI classifier without any target-user data
A2	N	Input	NFA-II	Reveals that some brain signal classes are significantly easier to forge than others
A3	N	Input	NFA-III	Forgery transfers across subjects and recording devices with no retraining
A4	E	Input	TDA-Truncation	Sub-second timing shift, injecting no extra data, degrades classifier to chance
A5	E	Input	TDA-Wrap	Cyclic time shift preserves signal power, bypassing energy-based anomaly detectors
A6	R	Transport	NRA-Raw	A recorded brain signal replayed over the wireless link hijacks BCI commands
A7	R	Transport	NRA-Augmented	Frequency-perturbed replay defeats statistical fingerprinting while preserving effect
A8	E_{bd}	Model	BEB	Hidden trigger disguised as a natural eye-blink artifact
A9	E_{bd}	Model	RPP	Trigger mimics electrical interference; leaves no trace in clean-class outputs
A10	E_{bd}	Model	CS	Trigger resembles a slow brain oscillation; invisible to standard quality checks
A11	E_{bd}	Model	DS	Trigger mimics muscle artifact; zero clean-class leakage
A12	E_{bd}	Model	TS	Structured transient trigger indistinguishable from benign neural activity
A13	E_{bd}	Model	OB	Movement-like noise trigger; trades stealth for higher raw success rate
A14	E_{bd}	Model	TP	Rhythmic artifact trigger; zero clean-class leakage
A15	E_{bd}	Model	SP	Mechanical noise pattern used as covert backdoor trigger
A16	E_{bd}	Model	ARC	Asymmetric discharge shape; undetectable by class-wise error monitors
A17	E_{bd}	Model	SWP	Trigger blends into background brain fluctuations; evades amplitude-based detection

tor for adversarial EEG payloads, bridging the GenAI and BCI security communities. Unlike prior attack taxonomies [18], [54] and fragmented BCI security analyses [12], [39], [40], NERVE Attacks span five orthogonal dimensions from physical transport to trained model, and EEGle is the first extensible security analysis platform for AI-powered BCIs, offering a foundation that can be used by other human-centred wearable AI systems.

III. Threat Model

Attacker goal. We assume an adversary aims to compromise a user's cognitive autonomy, mental privacy, or physical safety by subverting the AI-driven classification pipeline of a BCI system—either by manipulating model inputs or outputs in real time or by persistently corrupting and compromising the underlying model or infrastructure.

Attacker capabilities. We model an attacker to start *without* root, OS-level, or network privileges and *without* physical access to the target device. The attacker may occupy one of five entry points (E0–E4 below), each representing a distinct initial foothold requiring a different level of effort and infrastructure access. From any entry point, the attacker exploits the access-control and protocol weaknesses catalogued in Section IV to escalate capability. We do *not* assume neuroscience expertise: as we show later, a capable LLM can serve as a constrained domain-knowledge proxy, lowering the barrier to entry across all five NERVE dimensions. Physical side-channel attacks and OS-kernel exploits are explicitly out of scope.

Entry points (E0–E4). We define five attack entry points ordered from most to least capability required. Escalation across entry points exploits four *Vein Tapping* weaknesses (V1–V4, formally defined in Section IV): absent BLE link-layer encryption (V1), absent MAC-address authentication (V2), inadequate access control on BCI devices, data, sockets, and model files (V3), and insecure BCI SDKs and tooling implementations (V4).

E0 – API / Update-Channel Compromise. The attacker may control or impersonate a cloud or service provider endpoint in the BCI model-delivery path (e.g., via DNS spoofing or a malicious CI/CD workflow in the BrainFlow repository), enabling silent mass distribution of backdoored models to all connected devices without any local presence.

E1 – Third-Party Libraries and Supply-Chain Compromise. The attacker might poison a package registry entry consumed by the BCI stack (e.g., PyPI typosquatting `brainflow`, or dependency confusion on a private `mne-python` fork), executing arbitrary code at the BCI application's own privilege level on install or import.

E2 – Co-located Unprivileged Process. The attacker controls an ordinary userspace process or existing application on the BCI host and exploits absent access controls to read world-readable classifier files (exposing model weights) and overwrite world-writable model paths (injecting a backdoor), all without elevated privilege or network activity.

E3 – Network-Adjacent. An attacker could attempt to compromise or gain unauthorized access from the same Wi-Fi network or LAN. This foothold presumes that BCI host services expose the live neural stream on locally reachable sockets without authentication; we confirm that precondition empirically for the three evaluated platforms in Section VII-C.

E4 – Radio-Adjacent. An attacker within BLE range can passively sniff the unencrypted neural stream (V1); MAC-address spoofing escalates this to an active man-in-the-middle position for real-time payload injection (V2), as we demonstrate on several existing BCI devices. Privilege escalation from radio-adjacent observer to root-level code execution via

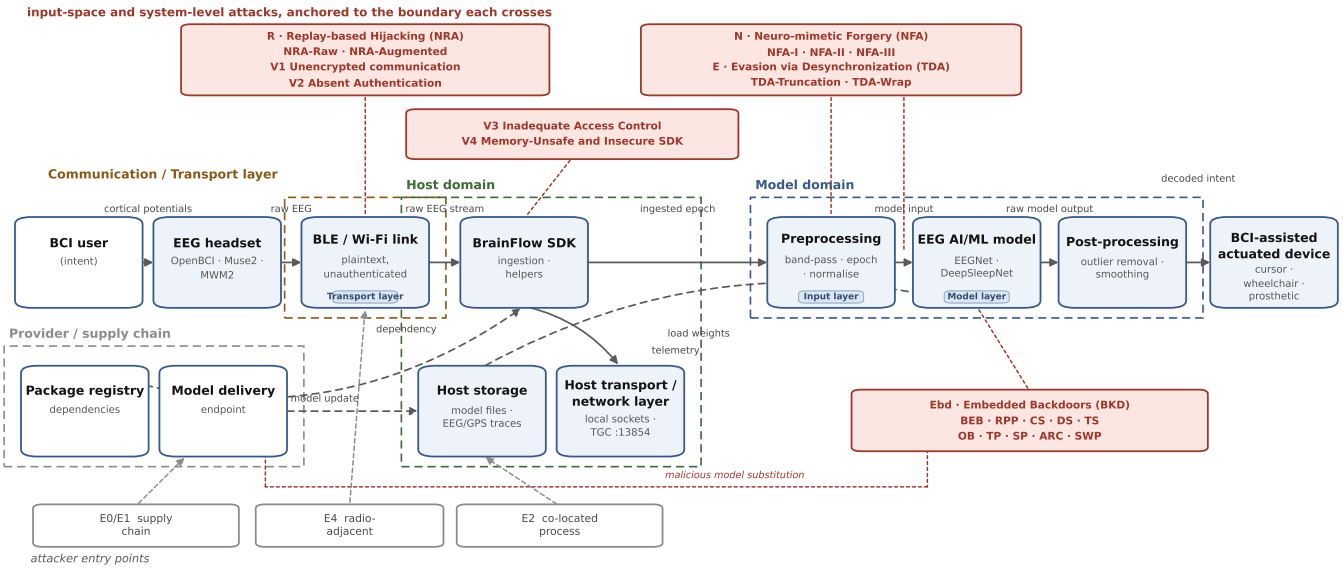


Fig. 1: The AI-powered BCI stack and the NERVE dimension that targets each layer. Each dimension is defined at a distinct trust boundary.

the BrainFlow confused-deputy path has been confirmed on real hardware [12], and Section VII-C verifies that every implicated weakness remains unpatched.

System assumptions. We assume a standard consumer BCI deployment: a wireless headset (OpenBCI, Muse, or NeuroSky) communicating via BLE or Wi-Fi to a host running a BrainFlow-based application with cloud or other service-provider connectivity for model updates and inference. We assume configuring the system with all security features available such as BLE link-layer encryption, MAC-address allowlisting, per-application authentication on BCI data sockets, model integrity signing, and memory-safe SDK APIs. We assume the host is a smartphone, or another execution environment shared with other applications, which makes co-located adversaries realistic.

Attacker knowledge. We assume a *gray-box* adversary whose prior knowledge is calibrated per NERVE dimension. For **N** (Neuro-mimetic Forgery): paradigm class, spectral bands, and electrode placement are publicly documented [39], [40]; an LLM generates the required physiological parameters without specialist expertise. For **E** (Evasion via Desynchronisation): the temporal structure of the stimulus-locked processing pipeline is known, enabling desynchronising perturbations that survive standard preprocessing. For **R** (Replay-based Hijacking): prior stream access (E3 or E4) suffices to collect a reference epoch corpus; no model knowledge is required. For **V** (Vein Tapping): BCI SDKs and API versions, port numbers, and filesystem paths are typically public, and the BLE re-pairing weaknesses the transport attacks build on are CVE-indexed [43]. The BrainFlow defects we report in Section VII-C are, by contrast, previously undisclosed and carry no CVE at the time of writing; they were reported to the maintainers under the disclosure process of Section IX. For **E**-backdoor (Embedded Backdoors): we assume model architecture is public (e.g., EEGNet [28]); weights are obtainable via unprotected readable files (E2) or from public repositories for foundation models that are used as a key building block such as BIOT, LaBraM, and EEGPT, making white-box attacks practical without even unauthorized access to model files on the host.

IV. The NERVE Attacks

Prior work has studied narrow, isolated BCI threat surfaces—adversarial examples [16], [17], replay attacks [12], or BCI software stack insecurity [35], [36]—without characterizing the modern AI-introduced attack surface or introducing the neuro-specific attack classes that this surface enables. We introduce the *NERVE Attacks*, named for its five orthogonal attack dimensions, each of which encompasses novel attack instances discovered in this work. Collectively, these dimensions cover the critical trust boundaries across the AI stack, from frameworks and trained models to the lower-level APIs and systems they depend on. This coverage surfaces novel attack vectors and establishes a principled foundation for systematic security evaluation.

A. Formal Definitions

We model a BCI system as the tuple $F = (I, \mathcal{E}, S, \Phi, V)$, whose five components are the system-state universe I, the extensible engine set $\mathcal{E}$, the orchestrator S that executes engines, the feedback component Φ that drives iteration, and the accumulated vulnerability set V. This is all that is needed to state the attack primitives below; Section V develops each component in full. We define five attack primitives over the components of F.

*a) **N** — Neuro-mimetic Forgery (NFA).:* An attacker $\mathcal{A}$ synthesises a class-c^* signal $\hat{x}$ from a public resting-state epoch x_{ref} (BCI Competition IV 2a) that is accepted as physiologically plausible and classified by the target model M as c^*:

$$\hat{x} \leftarrow \mathcal{A}(\text{paradigm}, c^*, x_{\text{ref}}), \quad M(\text{preprocess}(\hat{x})) = c^*. \quad (1)$$

The attacker bandpass-filters x_{ref} to isolate the mu (8–12 Hz) and beta (13–30 Hz) components and attenuates them at the channel associated with the desired class. Since artifacts, $1/f$ structure, and cross-channel covariance pass through unchanged, $\hat{x}$ is physiologically plausible by construction, without, for example, the use of a GAN. Relocating the suppression yields any other MI class from the same x_{ref}, so a single reference epoch produces arbitrarily many labelled samples. Large language models assisted in the design and implementation of the generation pipeline: $\hat{x}$ is produced by deterministic parametric code from x_{ref}, c^*, and a random seed. An attacker injects $\hat{x}$ into the live BLE stream via the Vein Tapping infrastructure V.

*b) **E** — Evasion via Desynchronization (TDA).:* An attacker introduces a controlled time shift δ to the input signal epoch:

$$x_\delta(t) = x(t - \delta), \quad \delta \in [\delta_{\min}, \delta_{\max}]. \quad (2)$$

Because EEG classifiers are trained on time-locked windows, even modest δ moves discriminative features outside the model's receptive field, causing accuracy to collapse. The attack is *query-free and payload-free*: no synthetic signal is required, only calculated clock offsets or BLE injection delay.

*c) **R** — Replay-based Hijacking (NRA).:* An attacker records an epoch x_c labelled as class c by the target model, then replays it verbatim or augmented via $x'_c = x_c \oplus \epsilon$ to elicit the same output:

$$M(\text{preprocess}(x'_c)) = c. \quad (3)$$

Augmentation ϵ is designed to preserve class-defining spectral features while defeating bytewise or template-matching replay detectors. Latency from start of replay to target-class detection is measured in seconds (quantified in Section VII-A).

*d) **V** — Vein Tapping.:* This includes the interception and manipulation of neural data at the BCI *infrastructure* and *application* layer—the wireless transport, authentication, and access-control mechanisms—before it reaches the AI model. The metaphor captures both the intimacy of the target (neural data as the body's "cognitive vein") and the attack's position: upstream of all ML defences. We identify four empirically verified attack surfaces (V1–V4):

- **V1 – Unencrypted communication:** All three evaluated devices (OpenBCI, Muse2, NeuroSky MindWave Mobile 2) transmit raw EEG and derivative metrics over plain-text, unencrypted channels. A passive attacker can sniff the data stream with commodity hardware.
- **V2 – Absent Authentication:** For instance, host applications fail to validate the BLE peripheral's MAC address. Hence, an attacker advertises an adversary-controlled device as the legitimate headset; the host connects without challenge.
- **V3 – Inadequate Access Control:** NeuroSky's ThinkGear Connector (TGC) forwards headset data over an unprotected TCP socket on port 13854, granting any local process access to brainwave data without permission. eegID stores EEG and GPS data in a world-readable CSV accessible to any app holding the coarse `STORAGE` permission.
- **V4 – Memory-Unsafe and Insecure SDK:** OpenBCI's BrainFlow SDK and UI show memory-corruption, race conditions, and insecure API logic, enabling arbitrary code execution, unauthorized access or elevated privilege.

Collectively, V1–V4 confirm that the access needed to exploit AI-layer vulnerabilities is not a theoretical assumption but an achievable precondition.

*e) **E** — Embedded Backdoors (BKD).:* An attacker with low-privilege host access replaces the clean model files M with a trojaned version M_τ that behaves identically on clean inputs but maps any trigger-overlaid input $x \oplus t_i$ to an attacker-chosen class c_i^*:

$$M_\tau(x) = M(x) \;\text{ if no trigger}, \quad M_\tau(x \oplus t_i) = c_i^* \;\forall\, i. \quad (4)$$

Persistence is key: once installed, M_τ activates silently on a covert external stimulus (e.g., a flicker at a specific frequency) with no further network activity. We embed $M{=}10$ distinct triggers simultaneously in EEGNet with an average ASR of 97.8% and negligible clean-accuracy cost (Section VII-B).

V. EEGle Overview

EEGle is designed for proactive, extensible security analysis of AI-powered BCI systems. Its dynamics are captured by $F = (I, \mathcal{E}, S, \Phi, V)$: I is the system-state universe; $\mathcal{E}$ the extensible engine set; S the SEO that executes engines; Φ the AI-assisted Feedback Engine that drives iteration; and $V = \bigcup_k R_k$ the accumulated vulnerability set. At each step k, the SEO executes a plan π_k selected by Φ, producing findings R_k, which update the state: $I_{k+1} = \Phi(I_k, R_k)$.

Security Engine Orchestration (SEO). The SEO is a stateless worker managing the registered engine set $\mathcal{E}_{reg}$. Its `execute_plan(`π_k`,` I_k`)` method identifies the engine e_i where e_i.name $= \pi_k$ and invokes `launch(`I_k`)`, returning R_k. A secondary `register_engine(`e_{new}`)` call lets Φ extend $\mathcal{E}_{reg}$ at runtime with LLM-synthesised engines.

The SEO operates without intrinsic knowledge of the overall analysis strategy. It neither interprets the findings nor makes decisions regarding the subsequent engine selection. Its role is strictly limited to the execution of the directives issued by Φ. A crucial secondary function, `register_engine(e_new)`, enables Φ to dynamically extend the set of registered engines $\mathcal{E}_{reg}$ during the analysis runtime, for example, by adding newly synthesized engines generated through LLM interaction. This dynamic registration capability is fundamental to the framework's adaptability and its capacity to explore novel attack vectors.

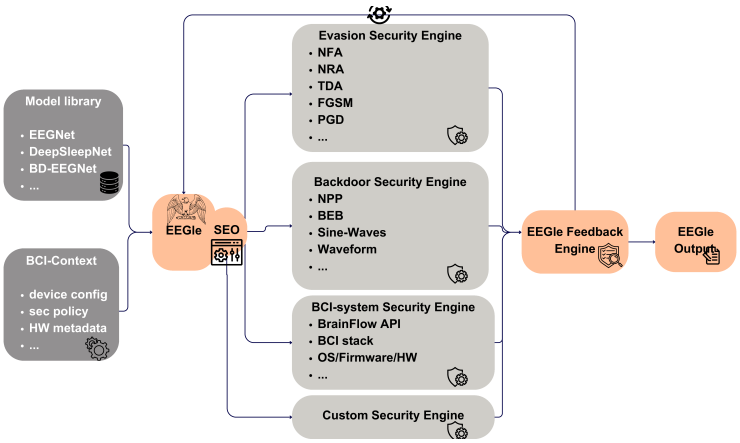


Fig. 2: EEGle high-level architecture. Given a system state I_k and a plan π_k, the SEO dispatches a registered Security Engine (evasion, backdoor, or system-level). Findings R_k update the state and feed the AI-assisted Feedback Engine Φ, which selects the next engine—enabling extensible, model-agnostic BCI security exploration.

Foundational Security Engines. The set of Security Engines $\mathcal{E} = \{e_1, e_2, \ldots, e_n\}$ constitutes the operational core of the framework, representing the diverse repertoire of actions that can be performed during the analysis. Each engine $e_i \in \mathcal{E}$ is formally defined as a function that maps the current system state I_k to a specific result or finding R_k ($e_i : I_k \rightarrow R_k$). Engines share a single narrow interface, which is what makes the repertoire extensible: a new engine is admissible as soon as it implements that interface, so Φ can register one at runtime without any change to the orchestrator. The collection $\mathcal{E}$ is further categorised by the engine's primary function to facilitate structured analysis.

For our evaluation, three built-in engines cover all 5 NERVE dimensions: the `EvasionEngine` (e_{eva}) for N/E/R attacks; the `BackdoorEngine` (e_{bd}) for E (Embedded Backdoors); and the `SystemEngine` (e_{sys}) for V (Vein Tapping). Custom engines can be registered for extended Red-Team/Blue-Team scenarios.

AI-Assisted Feedback Engine (Φ). This is EEGle's stateful control component and the architectural element that distinguishes it from a static scanner. At each iteration it receives the full result set R_k produced by the SEO alongside two persistent contextual inputs—the active security policy and the BCI context file encoding the current execution environment—and computes the successor state via $I_{k+1} = \Phi(I_k, R_k)$. Crucially, this computation is not a shallow aggregation of R_k in isolation; Φ evaluates each result in direct relation to the overall system state I_k, giving it the cross-iteration memory necessary to detect subtle compound vulnerabilities that no single-pass engine could surface.

The primary output of Φ is the next plan π_{k+1}, a structured specification that names the security engine to execute, its parameterisation, and the sub-objectives it should pursue. To produce π_{k+1}, Φ performs two logically distinct operations. First, *strategic reasoning*: it cross-references ML-level attack results with system-level findings across the full NERVE surface, identifies coverage gaps, and decides which attack dimension to probe next. This reasoning is performed by the LLM acting as a constrained planner whose output space is bounded by the registered engine repertoire and the security policy—the LLM cannot emit arbitrary actions, only valid plans. Second, *generative extension*: when the static engine repertoire is exhausted or a novel attack surface is identified, Φ invokes the LLM in a generative role to emit a structured JSON specification for a new payload or engine class. This specification is immediately validated, registered with the SEO via `register_engine`, and executed in the same iteration, so the framework's coverage expands at runtime without human intervention.

This separation of strategic reasoning from generative capability is a deliberate design choice. Conflating the two would allow unconstrained LLM output to drive execution, introducing brittleness and unpredictability. By keeping the LLM in a *specification* role rather than an *execution* role, Φ maintains deterministic auditability: every engine that runs can be traced to a validated plan, and every plan can be traced to a specific system-state transition. The closed loop ensures EEGle's coverage scales with the evolving BCI attack surface rather than being bounded by pre-defined templates.

VI. Implementation

A. Target Ecosystem & Assumptions

EEGle operates against a concrete target ecosystem: a consumer BCI deployment in which a wireless headset (OpenBCI Cyton, Muse2, or NeuroSky MindWave Mobile 2) communicates via BLE or Wi-Fi to a host running a BrainFlow-based application with cloud connectivity for model updates and inference. The target is specified through two JSON artefacts loaded at startup: *security_policy.json* enumerates the attack dimensions to exercise and their parameter bounds; *bci_context.json* catalogues the hardware identifiers, BrainFlow version, SDK paths, targetable ML models, and known service endpoints. Together these artefacts constitute EEGle's threat database and asset universe. Because BrainFlow exposes

a device-agnostic API, the SEO uses the same engine interface for real hardware and the fully emulated Synthetic Board (used for reproducible backdoor and NFA benchmarking). This abstraction ensures that results from the emulated board are directly portable to physical hardware without engine modification.

B. Evasion Security Engine

The engine probes BCI models under two regimes. In the *black-box* setting (no gradients), it deploys three BCI-specific attacks (NFA, NRA, TDA). When gradients are available, it additionally runs white-box robustness probes via standard ART methods [55] (FGSM, PGD, C&W, DeepFool), adapted for EEG preprocessing pipelines; formal definitions are in Appendix E.

We model an EEG window as $\mathbf{X} \in \mathbb{R}^{C\times T}$ and distinguish three injection placements: (i) sensor/transport-space (upstream of P_{ext}), (ii) preprocessing-space (midstream), and (iii) model-space (downstream). Additionally, we enable Expectation Over Transformations (EOT) when the injection is upstream of stochastic preprocessing:

$$\mathbb{E}_{\tau\sim\mathcal{T}}\big[\, L(f(\tau(\cdot)),\cdot)\,\big].$$

1) Black-Box Evasion Attacks:

- **Neuro-mimetic Forgery Attacks (NFA)** injects synthetic, physiologically plausible EEG epochs to control model outputs based on a generic time window of neural activity from a public dataset [56], [57].
  The attacker takes a resting-state epoch x_{ref} and imposes class-specific event-related desynchronization (ERD) to fabricate a synthetic motor imagery trial. Concretely, the mu (8–12 Hz) and beta (13–30 Hz) components of x_{ref} are extracted via bandpass filtering, attenuated by suppression factors $D_\mu = 0.90$ and $D_\beta = 0.70$ respectively, and subtracted from the original epoch at a spatially localised region around the class-appropriate focal electrode: C4 for left hand MI (contralateral right-hemisphere ERD), C3 for right hand MI (contralateral left-hemisphere ERD), and Cz for feet MI (central midline ERD). Suppression falls off spatially as a Gaussian ($\sigma = 0.40$) over 2D scalp distance from the focal electrode.
  Mental-state synthesis emulates relaxation, focus, stress, and drowsiness via canonical band-power ratios ($\delta/\theta/\alpha/\beta/\gamma$) with $1/f$ noise. Full component equations are in Appendix C.
- **Neuro Replay Attacks (NRA)** is a practical, black-box technique requiring no model internals. An attacker sniffs or MitMs the BLE stream (enabled by NERVE-V), records epochs and their classifier outputs, then builds an exemplar database. Replaying any entry reliably elicits the corresponding output. To defeat simple replay checkers, frequency-domain augmentation preserves class-defining spectral features while introducing variability:

$$\mathbf{X}^{adv} \;=\; (1-\lambda)\,\mathbf{X} \;+\; \lambda\,\text{EEG}_{\text{syn}}, \qquad \lambda \in [0,1].$$

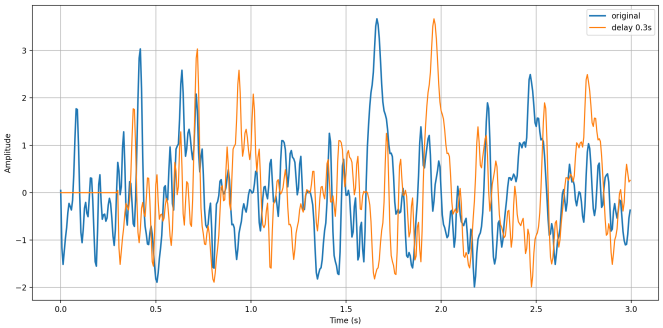


Fig. 3: EEG segment before (blue) and after a 0.3 s temporal shift attack (orange); early values are replaced by the channel mean.

If injected upstream of stochastic preprocessing, robust success is assessed via EOT over $f(\tau(\mathbf{X}^{adv}))$, $\tau \sim \mathcal{T}$. Full augmentation pipeline details are in Appendix D.

- **Temporal Desynchronization Attacks (TDA)** exploits the assumption of perfect temporal alignment inherent in event-locked paradigms (P300, SSVEP, MI). A controlled time shift τ moves class-defining features outside the model's receptive field, causing classification to fail with zero synthetic payload. We consider two variants (Figure 3), which differ in what they do with the vacated leading samples and therefore in what an anomaly detector can observe. Variant *(a)*, *truncation-shift*, delays the epoch and back-fills the first τ samples with the channel mean; this is trivial to mount at the transport layer but lowers the epoch's total signal power, which an energy-based detector could in principle flag. Variant *(b)*, *cyclic wrap*, instead rotates the epoch so that the displaced samples reappear at the end; total power and the amplitude distribution are preserved exactly, so it defeats energy- and variance-based checks at the cost of a discontinuity at the wrap point:

$$\textit{(a)}\;\; \mathbf{X}^{adv}(t) = \begin{cases} \text{mean}(\mathbf{X}), & t < \tau, \\ \mathbf{X}(t-\tau), & t \geq \tau, \end{cases}$$

$$\textit{(b)}\;\; \mathbf{X}^{adv}(t) = \mathbf{X}\big((t-\tau) \bmod T\big).$$

Both are computationally trivial to deploy at the transport layer.

C. Backdoor Attack Engine

Backdoors are *trained-in associations* between small, structured waveforms and attacker-chosen outputs, leaving clean accuracy intact while enabling reliable test-time activation. The engine exposes two phases—*implant* (poisoned fine-tuning) and *activate* (trigger overlay)—and a plan specifies the trigger family, target code, and amplitude.

We instantiate a *multi-trigger* backdoor with codebook $\mathcal{C} = \{t_i\}_{i=1}^{M}$ using transfer learning on EEGNet and DeepSleepNet. Each trigger maps one-to-one to a target class via

TABLE II: Backdoor trigger codebook: ten artifact-shaped templates embedded into EEGNet ($M = 10$) and five into DeepSleepNet ($M = 5$). *Clean Target rate* is the fraction of *clean* (un-triggered) samples that the backdoored model already assigns to the trigger's target class; a low rate means any observed misclassification is trigger-driven rather than pre-existing bias. Stealth tier follows directly from it: High $<5\%$, Moderate 5–40%, Lower $>40\%$. Per-trigger values are reported in Table VII.

Abbrev.	Waveform Structure	Resembles	Stealth
BEB	Gaussian deflection + negative undershoot, 1 Hz	Eye blink	Moderate
RPP	Short rectangular pulses, periodic	Digital interference	High
CS	Low-amplitude 3 Hz sine, full segment	Slow oscillation	High
DS	Paired narrow spikes, ∼0.5 s apart	Muscle transient	High
TS	Triple consecutive spikes, periodic	Structured transient	High
OB	15 Hz burst, Hann-windowed onset/offset	Movement artefact	Lower
TP	Repeating triangular waves	Rhythmic artefact	High
SP	Sawtooth ramp-up + sharp drop	Mechanical noise	High
ARC	Gradual ramp + fast decay + undershoot	Asymmetric transient	High
SWP	Exponential rise and decay	Background fluctuation	Lower

$\sigma(i) = y_i^\star$. Poisoning overlays a low-amplitude waveform: $X \mapsto X \oplus \gamma\, t_i$, with γ small to remain visually unobtrusive. Full hyperparameters are in Appendix H.

D. BCI-System Security Engine

This engine identifies the infrastructure vulnerabilities constituting NERVE-V (Vein Tapping). Its logic is adaptive: when source-code paths are provided, it runs a hybrid *static analysis*, combining traditional tooling with LLM-assisted code review, scanning for cryptographic failures (absent encryption, hardcoded keys), unsafe networking (open unauthenticated ports), access-control weaknesses (world-readable storage), and unsafe C memory operations (`memcpy`/`strcpy`) in BrainFlow and firmware components. When source code is unavailable, the engine shifts to *dynamic runtime analysis*, passively sniffing BLE packets to detect plain-text telemetry and automating MitM simulations to verify whether the host application accepts data from an unauthenticated source.

Using this engine, our analysis surfaced over 100 candidate memory- and API-security weaknesses across the stack. These are static-analyser *candidates*, not confirmed exploitable vulnerabilities: BrainFlow alone yielded 34 externally controlled format-string sites and 317 potential memory-corruption sites, which we treat as an upper bound on the surface available for privilege escalation and triage in Section VII-C. These findings map to the V1–V4 attack surfaces described in Section IV.

E. Feedback Engine

The Feedback Engine (Φ) is a stateful Python class that enforces the formal state transition $I_{k+1} = \Phi(I_k, R_k)$. It separates *strategic intelligence*, cross-referencing system-level findings with ML attack success, from the *generative capability* of an external LLM API call. When the static attack repertoire is exhausted, Φ uses the LLM as a highly constrained parameter generator (structured JSON output) to synthesise novel attack specifications, which are then registered as new engines in the SEO.

Fig. 4: CYBATHLON 2024 BCI game tasks [58]: Cursor control, Wheelchair navigation, and Ice Machine manipulation. All three were successfully hijacked in our end-to-end NRA/NFA demonstration. Misclassification in the Ice Machine task causes the robotic arm to tip, illustrating the physical safety stakes.

VII. Evaluation

Goals. We evaluate EEGle and the NERVE Attacks class against the following questions: *(1)* Does EEGle find emergent and novel threats across all five NERVE dimensions on modern AI-powered BCI platforms (summarized in Table I)? *(2)* How does each Security Engine perform in terms of attack efficacy? *(3)* How efficient is EEGle for developers compared to alternative approaches? Hardware, software, and device setup are detailed in Appendix G. Table III summarises which platform, model, and dataset each NERVE dimension was evaluated on, so that every result below can be traced to the component of the BCI stack it affects.

A. NERVE-N/E/R: Evasion Security Engine

1) Black-Box Evasion Attacks:

NFA. NFA demonstrated targeted control across all evaluated models (BrainFlow built-ins and EEGNetv4 MI classifier). LLM-assisted synthesis achieved targeted control of BrainFlow mental-state classifiers within 10 iterations. Against the MI victim classifier, the method exceeded chance (33.3%) across all tested conditions (Table IV): same-subject rest epochs achieved up to 61.0% overall accuracy (Subject 9), while cross-subject and cross-device conditions remained effective (44.7–52.3%), confirming that the attacker needs neither target-user data nor matched recording hardware. Per-class results reveal consistent asymmetry: certain classes are substantially easier to forge (e.g. right-hand MI reaches 83% for Subject 9 same-subject, while feet MI drops to 33%), consistent with known difficulty differences across MI classes reported in the motor-imagery decoding literature [27], [56].

NRA. To quantify realistic attacker effort, we measured the wall-clock time to identify one epoch per target class during 12 replay sessions (Table V). All target classes were reachable in a median of 7.3 s, with long-tail outliers attributable to random epoch ordering. Frequency-domain augmentation of replayed epochs preserved class-defining spectral features, defeating bytewise and simple template-matching replay checkers. More advanced statistical-fingerprint checkers are also subverted, since our augmentation preserves cross-channel covariance while introducing variability; robust detection requires multimodal defences (cryptographic channel protection, device au-

TABLE III: Evaluation platform by NERVE dimension. Each row states the target component, the model or device under test, and the data source, so that each result can be attributed to a specific part of the BCI stack. Full hardware and software versions are in Appendix G.

Dim.	Target component	Model / device under test	Data source	Reported in
N	Preprocessing input	EEGNet (BCI-IV-2a); BrainFlow classifiers	BCI-IV-2a, NeuroTUM	Sec. VII-A, Table IV
E	Preprocessing input	EEGNet (MMI); DeepSleepNet	PhysioNet MMI, Sleep-EDF	Sec. VII-A, Fig. 5
R	BLE transport	EEGNetv4 + CYBATHLON emulator	BCI-IV-2a replay corpus	Sec. VII-A, Table V
V	Transport and host	OpenBCI Cyton, Muse2, NeuroSky MWM2; BrainFlow 5.18.0	Live device capture	Sec. VII-C
E_{bd}	Trained model	EEGNet (MMI); DeepSleepNet	PhysioNet MMI, Sleep-EDF	Sec. VII-B, Table VII

TABLE IV: Classification accuracy (%) of synthetic v2 signals evaluated on EEGNet trained with real BCI-IV-2a [56] data. Rest sources include same-subject and cross-subject BCI-IV-2a fixation epochs, and NeuroTUM recordings [57]. Chance level: 33.3%.

Rest source	Subj.	Feet	Left	Right	Overall
Real baseline	3	75.0	83.3	91.7	**83.3**
	9	58.3	95.8	75.0	**76.4**
Same-subj. BCI-IV	3	55	35	54	**48.0**
	9	33	67	83	**61.0**
Diff.-subj. (S1) BCI-IV	3	22	42	74	**46.0**
	9	20	81	47	**49.3**
NeuroTUM rest	3	77	24	56	**52.3**
	9	34	67	33	**44.7**

TABLE V: NRA latency statistics: time to discover target-class epoch ($N = 12$ runs).

Metric	Value (s)
Mean	12.255
Median	7.345
Standard deviation	15.281
Minimum	1.270
Maximum	59.386

thentication, and liveness checks). We conducted an end-to-end demonstration using the three-class MI paradigm, the EEGNet classifier, and the CYBATHLON 2024 BCI game [58] (Figure 4). We evaluated three injection modalities: raw epoch replay, LLM-synthesised epoch injection, and augmented-replay; each successfully caused the emulator to execute adversary-chosen actions, confirming end-to-end exploitability.

TDA. Figure 5 shows mean classification accuracy under incremental time-shifts ($\delta = k \cdot T/10$), averaged over N=5 independent runs per step with ± 1 standard deviation error bands. EEGNet retains full accuracy for shifts up to $3T/10$ and then degrades sharply: 0.99 at $3T/10$, 0.88 at $4T/10$, 0.60 at $5T/10$, and 0.10 by $8T/10$ (std <0.04). Because the MMI classifier is a ten-class model, 0.10 is exactly chance level, so the shifted input carries no recoverable class information at all; the sharp, monotone descent confirms the effect is structural rather than noise. The minimum effective shift is $\delta_{\min} = 4T/10$ (≈1.2 s for a 3 s MI epoch); smaller shifts fall within the run-to-run standard deviation. A shift of this

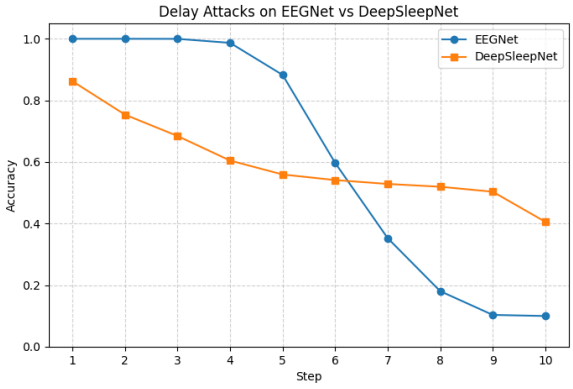


Fig. 5: Classification accuracy under Temporal Desynchronization Attacks. EEGNet (short-window, event-locked) collapses sharply; DeepSleepNet (long-window, stationary) degrades gradually but still loses half its baseline accuracy.

magnitude is achievable by a MitM relay that buffers and re-emits epochs (Section VII-C), though it is correspondingly easier to detect than a sub-second offset, and a compromised clock-synchronisation service remains the stealthier delivery path. DeepSleepNet degrades more gradually ($0.86 \rightarrow 0.41$, std up to 0.09), as its longer 30 s window partially absorbs small shifts. The attack is query-free and zero-payload: the only attacker capability required is a timing offset at the transport layer (E3/E4). Architecture-dependent degradation profiles confirm that temporal robustness must be probed per target and cannot be inferred from standard adversarial benchmarks; no published defence against TDA currently exists (Section VIII).

2) White-Box Robustness Probe: White-box attacks are used here as a *theoretical upper bound* on achievable attack success: an adversary usually lacks gradient access, but these results bound how much harder the black-box problem is. Table VI shows results using ART v1.20.1 [55]. EEGNet is markedly more vulnerable to L_∞-bounded perturbations (PGD: ASR = 1.000 at ϵ=0.2) than DeepSleepNet (PGD: ASR = 0.243 at the same budget), consistent with EEGNet's task-mismatch shortcut learning observed in the black-box experiments. Large-norm methods (DeepFool, $L_2 \approx 6711$ on DeepSleepNet) confirm that the model is not immune, but set a high cost for any practical adversary.

TABLE VI: White-box robustness probe (upper-bound). FGSM/PGD $\epsilon = 0.2$; PGD step size $\alpha = 0.1$ (`eps_step`); C&W 100 iter., with initial constant $c = 0.01$ (`initial_const`) and 10 binary-search steps (`binary_search_steps`); DeepFool 50 iter. No explicit random seed was set. *Clean* is accuracy on the unperturbed evaluation subset used for this probe; for DeepSleepNet this is 0.829, slightly below the ∼0.86 held-out accuracy reported in Appendix A-B and used as the TDA baseline in Figure 5, because the probe is run on a gradient-accessible subset rather than the full held-out set.

Attack	Clean	Adv.	ASR	L_2	L_∞
EEGNet					
FGSM	1.000	0.137	0.863	35.05	0.200
PGD	1.000	0.000	1.000	30.05	0.200
C&W	1.000	0.622	0.378	0.594	0.031
DeepFool	1.000	0.271	0.729	2233.3	37.24
DeepSleepNet					
FGSM	0.829	0.784	0.217	10.95	0.200
PGD	0.829	0.757	0.243	10.09	0.200
C&W	0.829	0.980	0.020	18.32	1.720
DeepFool	0.829	0.440	0.560	6711.3	222.8

TABLE VII: Multi-trigger backdoor evaluation. Clean Target = fraction of clean samples already predicted as target; ASR = attack success rate on trigger-overlaid inputs. EEGNet evaluated on held-out subjects S021–S030.

Trig.	Tgt.	Clean Tgt. (%)	ASR (%)	Stealth
EEGNet				
BEB	1	3.76	100.00	Mod.
RPP	2	0.00	99.06	High
CS	3	0.75	100.00	High
DS	4	0.00	100.00	High
TS	5	0.00	100.00	High
OB	6	59.21	100.00	Lower
TP	7	0.00	100.00	High
SP	8	0.00	78.95	High
ARC	9	0.00	100.00	High
SWP	10	47.18	100.00	Lower
Avg	–	**11.10**	**97.80**	
DeepSleepNet				
BEB	W	26.5	26.0	–
DS	N1	7.5	15.0	–
OB	N2	27.5	27.0	–
TP	N3	19.0	28.5	–
SWP	REM	19.5	3.5	–
Avg	–	**20.0**	**20.0**	

B. NERVE-E: Backdoor Attacks Engine

The PhysioNet MMI cohort is partitioned once and used consistently throughout: S001–S010 train the clean baseline, S011–S020 supply the poisoning pool, and S021–S030 are held out and never seen during training or poisoning, so the attack success and clean-accuracy figures in Table VII are measured on subjects disjoint from both. For EEGNet we poisoned subjects S011–S020, embedding $M{=}10$ triggers with one-to-one target mappings. For DeepSleepNet we injected five triggers mapped to canonical sleep stages (BEB→W, DS→N1, OB→N2, TP→N3, SWP→REM). Both models use the overlay policy $X \mapsto X \oplus \gamma\, t_i$ with low amplitude γ.

EEGNet results. All ten triggers achieve high ASR (avg 97.8%, Table VII). Nine reach saturation ($\geq$99%) and only SP falls below it (78.95%), confirming reliable multi-target control. Low average Clean Target (11.1%) shows that misclassification is backdoor-driven, not a pre-existing bias.

DeepSleepNet results. The same codebook and overlay produce an average ASR of only 20.0%, near or below the Clean Target rate for most triggers—indicating the model does not route triggered inputs to attacker-chosen stages. The architecture-dependent gap confirms that backdoor risk cannot be assessed without model-specific evaluation.

Stealth-Effectiveness Spectrum. The per-trigger results reveal a *stealth-effectiveness spectrum* fundamental to BCI backdoor design. We assign tiers by Clean Target rate, the fraction of *clean* samples already predicted as the trigger's target class: High for $<5\%$, Moderate for 5–40%, and Lower for $>40\%$. The *high-stealth tier* (RPP, CS, DS, TS, TP, SP, ARC: $\leq 0.75\%$ Clean Target) leaves no accuracy footprint, defeating post-hoc audits; six of these seven reach 99–100% ASR, with SP the single exception at 78.95%. BEB sits in the moderate tier (3.76% Clean Target, 100% ASR). The *lower-stealth tier* (OB: 59.2%, SWP: 47.2% Clean Target) is marginally detectable in class-wise error patterns, yet still reaches 100% ASR. An attacker can *choose* position on this spectrum—invisible implant vs. guaranteed activation—a trade-off absent in image-domain backdoor literature and unique to the physiological signal domain.

C. NERVE-V: Vein Tapping Results

We evaluated PoC attack vectors on the BrainFlow Emulator and three real-world platforms (OpenBCI, Muse, NeuroSky), empirically confirming all four V-dimension attack surfaces (V1–V4).

V1 – Passive sniffing. Using an nRF52840 dongle and Wireshark, we passively captured plain-text BLE frames on all three devices, recovering raw EEG and derivative metrics (e.g., "attention" at characteristic 0x001C on MWM2) without any authentication. A custom GATT sniffing plugin displayed live neural data without the device owner's knowledge.

V2 – MitM via MAC bypass. Two Raspberry Pis running GATTacker mediated all BLE traffic: one connected to the headset, the other advertising as the headset. NeuroSky and Muse apps accepted the adversarial peripheral without MAC verification, giving us read/write access to the live stream—confirming that MitM is trivially achievable and not merely theoretical.

V3 – Access control failures. NeuroSky's ThinkGear Connector (TGC) forwarded brainwave data over an unauthenticated TCP socket (port 13854); any local process connected without permission and without user notification. We successfully impersonated TGC, sending false cognitive-state data to downstream applications. The OpenBCI WebSocket endpoint (`ws://localhost:10996`) is likewise unauthen-

ticated and cross-origin accessible, so any browser tab the user has open can read the live EEG stream. eegID stored EEG and GPS traces in a world-readable CSV (`eegIDRecord.csv`), leaking location data to any app holding the coarse `STORAGE` permission.

V4 – Insecure SDK. We analysed the BrainFlow 5.18.0 C/C++ sources with CodeQL 2.23.6 and Cppcheck 2.18.2, which reported 34 externally controlled format-string sites and 317 potential memory-corruption sites. These are analyser *candidates*, not confirmed vulnerabilities.

We did not develop a working control-flow hijack from these sites, and we therefore make no claim of arbitrary code execution. What we confirmed dynamically is narrower but concrete: the OpenBCI GUI invokes BrainFlow helper components without privilege separation, so code executing inside a helper inherits the launching process's privileges rather than a reduced set. Establishing whether the triaged sites are exploitable in practice is left to future work; all findings were reported to the maintainers (Section IX).

D. Feedback Engine

To validate that the AI-Assisted Feedback Engine Φ provides coverage beyond static sequential engine execution, we compare two configurations: *Static-EEGle*, which runs the three foundational engines once in a fixed order (Evasion $\rightarrow$ Backdoor $\rightarrow$ System) without LLM-driven replanning; and *EEGle*+Φ, which uses the full feedback loop (more details in Table IX). Φ is implemented against a single commercial LLM API accessed through a structured-output (JSON) interface. We did not run a controlled comparison across model families, so we make no claim that the chosen model is better suited to this task than any other; Φ treats the model as an interchangeable component behind a fixed plan schema. Sensitivity of EEGle's findings to the choice of LLM is an acknowledged limitation and a direction for future evaluation.

VIII. Potential Defenses

In this section we discuss effective countermeasures and identify open problems for mitigating NERVE attacks.

Neuro-mimetic Forgery.

Defending against neuro-mimetic forgery requires distinguishing physiologically plausible EEG from genuine, ongoing neural activity. *Liveness detection* addresses this distinction through stimulus-evoked ERPs with characteristic timing and morphology [39], complemented by anomaly detection on band-power ratios. On the model side, alignment-based adversarial training (ABAT [59]) improves resistance to perturbations, while EEG benchmarks [49] support assessment of accuracy–robustness trade-offs. Randomised smoothing further provides certified bounds under specified perturbation models [60], [61]. However, these model-level protections do not establish signal authenticity, while stimulus-response liveness checks introduce latency and interaction overhead. The practical challenge is therefore to verify genuine neural activity without compromising the responsiveness required by applications such as prosthetic control.

Evasion via Desynchronisation.

Overlapping window averaging, multi-scale temporal pooling, and stimulus-locked integrity checks that reject out-of-window epochs provide partial mitigation. However, TDA exploits the event-locking assumption fundamental to P300, SSVEP, and MI paradigms. *No principled validated defence against TDA currently exists*: widening the acceptance window re-opens the attack surface while narrowing it degrades benign accuracy.

Replay-based Hijacking.

Proper access-control mechanisms, including HMAC or rolling-nonce epoch authentication, can prevent replayed epochs from being accepted; for example, BLE session freshness via LE Secure Connections removes the transport-layer replay surface [42], [43]; challenge-response liveness probes close the remaining window. However, most effective defences require firmware changes that BCI vendors have not shipped; for some legacy hardware the surface is unsolvable without replacement.

Vein Tapping.

The V1–V4 weaknesses span every layer of the software stack, and effective defences at each layer are well understood in the broader systems security literature—secure transport protocols, application-layer access control, proper isolation [62], [63], hardware/software-based compartmentalization [64]–[66], memory-safe languages and fuzzing-based code analysis, and supply-chain provenance frameworks (SLSA [46], Sigstore [67], in-toto [47]) collectively address the full surface. Hardware-based security solutions such as confidential-computing enclaves (ARM TrustZone [68], CCA [69]) have also been proposed to protect AI models' integrity and confidentiality from co-located untrusted components. Though their hardware requirements and performance overheads remain impractical for legacy resource-constrained BCI devices and large EEG foundation models, modern AI-powered wearable architectures and smaller (or quantized) models can benefit from these solutions (particularly in hybrid edge-cloud architectures).

Despite this maturity, no BCI ecosystem project has adopted any of these practices. Two systemic factors explain this. First, there is no *security-by-design* culture in this highly security-sensitive space, as evidenced most plainly by the absence of basic access control, process isolation, and privilege separation at every layer of the stack—properties that have been standard in general-purpose OS and mobile security for decades. Second, the ecosystem is caught in a functionality race: as BCI moves from laboratory curiosity to commercial reality, with major technology companies integrating neural interfaces into spatial computing and consumer wearables [6], [11], the pressure to ship compelling applications leaves security consistently deprioritised. Both dynamics were already observable when many similar vulnerabilities were reported to the community years ago [12], [35], [37]–[39], [70]–[72]; the intervening years, and the arrival of AI-assisted security tooling that has lowered the adoption barrier further still, have produced no measurable change. The gap is not technical in this specific

space—it is cultural and structural.

Embedded Backdoors.

Proper model signing and attestation techniques can prevent silent replacement, though an E0 attacker can replace model and key simultaneously. For *detection*, Neural Cleanse [73] reverse-engineers triggers; STRIP [74] uses runtime prediction entropy; Activation Clustering [75] clusters final-layer activations; Spectral Signatures [76] and SPECTRE [77] use SVD and robust covariance estimation; spatial-spectral analysis [78] targets the EEG setting. For *mitigation*, Fine-Pruning [79] prunes backdoor neurons; NAD [80] erases associations via distillation on 5 % clean data; ANP [81] targets adversarially sensitive neurons; i-BAU [82] achieves comparable results with $\geq$100 samples.

However, recent work [83] shows backdoors *persist* after most such defences: a modified trigger re-activates them. Clean-label backdoors evade Activation Clustering and Neural Cleanse by design. Whether Fine-Pruning and NAD transfer to foundation models (BIOT, LaBraM, EEGPT) is unknown.

Design principles for BCI vendors. The matrix supports four concrete recommendations, ordered by the ratio of risk removed to engineering effort. *First*, enable BLE link-layer encryption and LE Secure Connections and verify the peripheral's identity at the host: this closes V1–V2 outright and removes the transport-layer replay surface (R) with no ML changes at all. *Second*, treat the neural data path as privileged: authenticate local sockets, drop world-readable storage of EEG and derived metrics, and run SDK helpers with reduced privilege. This closes V3 and contains V4 regardless of whether the underlying memory-safety defects are ever fixed. *Third*, sign models and verify the signature at load time, and pin the update endpoint. This raises backdoor implantation (E_{bd}) from a file overwrite to a key-compromise problem. *Fourth*, treat temporal alignment as a security property, not only a signal-processing one: validate epoch timing against an authenticated clock and reject epochs whose stimulus-locking cannot be confirmed. We stress that the fourth is the least mature: as noted above, no validated defence against TDA currently exists, and this is the clearest open problem the NERVE analysis exposes.

Limitations and defence-in-depth.

No single countermeasure eliminates all NERVE risks. Existing open problems identified above share a common asymmetry: attacker capability scales with the same AI tooling and foundation models that power modern BCI pipelines, while defensive efforts and guarantees remain bounded by loose software security practices, certified-robustness bounds, unresponsive hardware vendors, and ecosystem-wide absence of supply-chain hygiene. No combination of current techniques provides end-to-end protection against a determined NERVE attacker, and the gap is widening—not closing.

IX. Responsible Disclosure

We disclosed every finding in this paper to the affected parties before submission: the BrainFlow maintainers, and the BCI vendors OpenBCI, NeuroSky, and Muse. We are committed to collaborating with them on necessary mitigation efforts before the paper is published, and we withhold exploit code for the system-level findings until fixes are available. Separately, as detailed in this paper, EEGle uses generative AI strictly for security automation and synthetic data generation.

X. Conclusion

We introduced the *NERVE Attacks*, a systematic characterisation of five orthogonal attack dimensions spanning the complex attack surface of modern AI-powered BCI systems, and EEGle, the first extensible framework for BCI security analysis—one whose architecture generalises naturally to other human-centred AI-powered wearables. Evaluated on three real-world BCI platforms and diverse EEG pipelines, our results confirm two findings with broad implications. First, modern AI-powered BCI systems are uniquely exposed to domain-specific semantic attacks: 17 novel neuro-specific instances demonstrate that physiological plausibility constraints place these threats outside the reach of standard adversarial ML toolboxes, while LLM-assisted payload generation is rapidly collapsing the expertise barrier for non-expert attackers. Second, the BCI software stack is insecure at every level, with each layer independently exploitable, as we showed in our evaluation. We release EEGle to the community as a foundational tool to audit and protect these deeply personal devices.

TABLE VIII: Mapping of NERVE attack classes to candidate mitigations. • = effective against the class as evaluated here; ○ = partial, i.e. raises attacker cost or narrows the window but does not close the surface; – = ineffective or not applicable. *Cost* is the dominant deployment barrier rather than a monetary figure. No row is fully covered by a single mitigation, which is the central point of this section.

Class	**Layer**	**Crypto transport**	**Liveness / freshness**	**Robust training**	**Attest. / signing**	**Dominant cost**
N (Forgery)	Input	○	•	○	–	Liveness adds >100 ms latency
E (Desync.)	Input	–	○	–	–	No validated defence exists
R (Replay)	Transport	•	•	–	–	Requires vendor firmware update
V (Vein Tap.)	Transport	•	○	–	○	Well understood; simply not adopted
	Host	–	–	–	•	Needs OS-level privilege separation
E_{bd} (Backdoor)	Model	–	–	○	•	Attestation fails against an E0 attacker

REFERENCES

[1] P. Lakhan, N. Banluesombatkul, V. Changniam, R. Dhithijaiyratn, P. Leelaarporn, E. Boonchieng, S. Hompoonsup, and T. Wilaiprasitporn, "Consumer grade brain sensing for emotion recognition," *IEEE Sensors Journal*, vol. 19, no. 21, pp. 9896–9907, 2019.

[2] S. N. Abdulkader, A. Atia, and M.-S. M. Mostafa, "Brain computer interfacing: Applications and challenges," *Egyptian Informatics Journal*, vol. 16, no. 2, pp. 213–230, 2015.

[3] M. Fatima, M. Shafique, and Z. Khan, "Towards a low cost brain-computer interface for real time control of a 2 dof robotic arm," in *2015 International Conference on Emerging Technologies (ICET)*, 2015, pp. 1–6.

[4] B. H. Kim, M. Kim, and S. Jo, "Quadcopter flight control using a low-cost hybrid interface with EEG-based classification and eye tracking," *Computers in Biology and Medicine*, vol. 51, pp. 82–92, 2014.

[5] S. R. A. Jafri, T. Hamid, R. Mahmood, M. A. Alam, T. Rafi, M. Z. U. Haque, and M. W. Munir, "Wireless brain computer interface for smart home and medical system," *Wireless Personal Communications*, vol. 106, no. 4, pp. 2163–2177, 2019.

[6] Synchron, Inc. (2025, May) Synchron to achieve first native brain-computer interface integration with iPhone, iPad and Apple vision pro. Press release, https://www.biospace.com/press-releases/synchron-to-achieve-first-native-brain-computer-interface-integration-with-iphone-ipad-and-apple-vision-pro. 13 May 2025. Accessed: 27 August 2026.

[7] X. Zhang, T. Zhang, Y. Jiang, W. Zhang, Z. Lu, Y. Wang, and Q. Tao, "A novel brain-controlled prosthetic hand method integrating AR-SSVEP augmentation, asynchronous control, and machine vision assistance," *Heliyon*, vol. 10, no. 5, p. e26521, 2024.

[8] J. S. Brumberg, K. M. Pitt, A. Mantie-Kozlowski, and J. D. Burnison, "Brain–computer interfaces for augmentative and alternative communication: A tutorial," *American Journal of Speech-Language Pathology*, vol. 27, no. 1, pp. 1–12, 2018.

[9] N. Dong, Z. Wu, W. Zhang, G. Chen, and Z. Gao, "Intention-prioritized fuzzy fusion control for bci-based autonomous vehicles," *Biomedical Signal Processing and Control*, vol. 87, p. 105486, 2024.

[10] O. Hatem, O. Sheta, A. Abdelaziz, A. Ashraf, H. El Maleeh, M. Ali, M. Gadallah, M. Habash, A. Moro, and S. Eldawlatly, "Brain-brake: A hybrid brain-operated computer vision-enabled system for collision avoidance," in *2024 IEEE 20th International Conference on Body Sensor Networks (BSN)*, 2024, pp. 1–4.

[11] S. Cheng, "The future of the metaverse," in *Metaverse: Concept, Content and Context*. Springer, 2023, pp. 207–215.

[12] Z. Tarkhani, L. Qendro, M. O. Brown, O. Hill, C. Mascolo, and A. Madhavapeddy, "Enhancing the security & privacy of wearable brain-computer interfaces," *arXiv preprint arXiv:2201.07711*, 2022.

[13] F. Mo, Z. Tarkhani, and H. Haddadi, "Machine learning with confidential computing: A systematization of knowledge," *ACM Computing Surveys*, vol. 56, no. 11, pp. 281:1–281:40, 2024.

[14] T. Lahtinen, A. Costin, and G. Suarez-Tangil, "Brain-computer interface integration with extended reality (XR): Future, privacy and security outlook," in *Proceedings of the 23rd European Conference on Cyber Warfare and Security (ECCWS)*, vol. 23, no. 1, 2024, pp. 265–271.

[15] X. Wang, M. Hersche, O. R. Q. Siller, L. Benini, and G. Singh, "Physically-constrained adversarial attacks on brain-machine interfaces," in *NeurIPS 2022 Workshop on Trustworthy and Socially Responsible Machine Learning (TSRML)*, 2022. [Online]. Available: https://openreview.net/forum?id=oogi4S33q8

[16] D. Wu, J. Xu, W. Fang, Y. Zhang, L. Yang, X. Xu, H. Luo, and X. Yu, "Adversarial attacks and defenses in physiological computing: A systematic review," *National Science Open*, vol. 2, no. 1, p. 20220023, 2023, preprint: arXiv:2102.02729.

[17] X. Zhang, D. Wu, L. Ding, H. Luo, C.-T. Lin, T.-P. Jung, and R. Chavarriaga, "Tiny noise, big mistakes: adversarial perturbations induce errors in brain–computer interface spellers," *National Science Review*, vol. 8, no. 4, p. nwaa233, 2021.

[18] M. Sharif, S. Bhagavatula, L. Bauer, and M. K. Reiter, "A general framework for adversarial examples with objectives," *ACM Transactions on Privacy and Security*, vol. 22, no. 3, pp. 16:1–16:30, 2019.

[19] S. Eldawlatly, "On the role of generative artificial intelligence in the development of brain-computer interfaces," *BMC Biomedical Engineering*, vol. 6, no. 1, p. 4, 2024.

[20] A. Raza and M. Z. Yusoff, "Deep learning approaches for EEG-motor imagery-based BCIs: Current models, generalization challenges, and emerging trends," *IEEE Access*, vol. 13, pp. 151 866–151 893, 2025.

[21] A. G. Habashi, A. M. Azab, S. Eldawlatly, and G. M. Aly, "Generative adversarial networks in EEG analysis: an overview," *Journal of Neuro-Engineering and Rehabilitation*, vol. 20, no. 1, p. 40, 2023.

[22] D. Wen, B. Liang, Y. Zhou, H. Chen, and T.-P. Jung, "The current research of combining multi-modal brain-computer interfaces with virtual reality," *IEEE Journal of Biomedical and Health Informatics*, vol. 25, no. 9, pp. 3278–3287, 2021.

[23] P. Tai, P. Ding, F. Wang, A. Gong, T. Li, L. Zhao, L. Su, and Y. Fu, "Brain-computer interface paradigms and neural coding," *Frontiers in Neuroscience*, vol. 17, p. 1345961, 2024.

[24] L. F. Nicolas-Alonso and J. Gomez-Gil, "Brain computer interfaces, a review," *Sensors*, vol. 12, no. 2, pp. 1211–1279, 2012.

[25] T.-P. Jung, S. Makeig, C. Humphries, T.-W. Lee, M. J. McKeown, V. Iragui, and T. J. Sejnowski, "Removing electroencephalographic artifacts by blind source separation," *Psychophysiology*, vol. 37, no. 2, pp. 163–178, 2000.

[26] H. Ramoser, J. Müller-Gerking, and G. Pfurtscheller, "Optimal spatial filtering of single trial EEG during imagined hand movement," *IEEE Transactions on Rehabilitation Engineering*, vol. 8, no. 4, pp. 441–446, 2000.

[27] F. Lotte, L. Bougrain, A. Cichocki, M. Clerc, M. Congedo, A. Rakotomamonjy, and F. Yger, "A review of classification algorithms for EEG-based brain–computer interfaces: a 10 year update," *Journal of Neural Engineering*, vol. 15, no. 3, p. 031005, 2018.

[28] V. J. Lawhern, A. J. Solon, N. R. Waytowich, S. M. Gordon, C. P. Hung, and B. J. Lance, "EEGNet: a compact convolutional neural network for EEG-based brain–computer interfaces," *Journal of Neural Engineering*, vol. 15, no. 5, p. 056013, 2018.

[29] A. Supratak, H. Dong, C. Wu, and Y. Guo, "DeepSleepNet: A model for automatic sleep stage scoring based on raw single-channel EEG," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 25, no. 11, pp. 1998–2008, 2017.

[30] C. Yang, M. B. Westover, and J. Sun, "BIOT: Biosignal transformer for cross-data learning in the wild," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 36, 2023, pp. 78 240–78 260. [Online]. Available: https://papers.nips.cc/paper_files/paper/2023/hash/f6b30f3e2dd9cb53bbf2024402d02295-Abstract-Conference.html

[31] W.-B. Jiang, Y. Wang, B.-L. Lu, and D. Li, "NeuroLM: A universal multi-task foundation model for bridging the gap between language and EEG signals," in *The Thirteenth International Conference on Learning Representations (ICLR)*, 2025, arXiv:2409.00101.

[32] W.-B. Jiang, L.-M. Zhao, and B.-L. Lu, "Large brain model for learning generic representations with tremendous EEG data in BCI," in *The Twelfth International Conference on Learning Representations (ICLR)*, 2024. [Online]. Available: https://openreview.net/forum?id=QzTpTRVtrP

[33] G. Wang, W. Liu, Y. He, C. Xu, L. Ma, and H. Li, "EEGPT: Pretrained transformer for universal and reliable representation of EEG signals," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 37, 2024, pp. 39 249–39 280. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2024/hash/4540d267eeec4e5dbd9dae9448f0b739-Abstract-Conference.html

[34] J. Wang, S. Zhao, Z. Luo, Y. Zhou, H. Jiang, S. Li, T. Li, and G. Pan, "CBraMod: A criss-cross brain foundation model for EEG decoding," in *The Thirteenth International Conference on Learning Representations (ICLR)*, 2025, arXiv:2412.07236.

[35] T. Bonaci, R. Calo, and H. J. Chizeck, "App stores for the brain: Privacy & security in brain-computer interfaces," in *2014 IEEE International Symposium on Ethics in Science, Technology and Engineering*, 2014, pp. 1–7.

[36] H. Takabi, A. Bhalotiya, and M. Alohaly, "Brain computer interface (BCI) applications: Privacy threats and countermeasures," in *2016 IEEE 2nd International Conference on Collaboration and Internet Computing (CIC)*, 2016, pp. 102–111.

[37] S. L. Bernal, A. H. Celdrán, G. M. Pérez, M. T. Barros, and S. Balasubramaniam, "Security in brain-computer interfaces: State-of-the-art, opportunities, and future challenges," *ACM Computing Surveys*, vol. 54, no. 1, pp. 11:1–11:35, 2021.

[38] O. Landau, R. Puzis, and N. Nissim, "Mind your mind: EEG-based brain-computer interfaces and their security in cyber space," *ACM Computing Surveys*, vol. 53, no. 1, pp. 17:1–17:38, 2020.

[39] I. Martinovic, D. Davies, M. Frank, D. Perito, T. Ros, and D. Song, "On the feasibility of side-channel attacks with brain-computer interfaces," in *21st USENIX Security Symposium (USENIX Security 12)*, 2012, pp. 143–158. [Online]. Available: https://www.usenix.org/conference/usenixsecurity12/technical-sessions/presentation/martinovic

[40] M. Frank, T. Hwu, S. Jain, R. T. Knight, I. Martinovic, P. Mittal, D. Perito, I. Sluganovic, and D. Song, "Using EEG-based BCI devices to subliminally probe for private information," in *Proceedings of the 2017 Workshop on Privacy in the Electronic Society (WPES)*. ACM, 2017, pp. 133–136.

[41] S. Jasek, "GATTacking Bluetooth smart devices: Introducing a new BLE proxy tool," Black Hat USA, whitepaper, 2016. [Online]. Available: https://blackhat.com/docs/us-16/materials/us-16-Jasek-GATTacking-Bluetooth-Smart-Devices-Introducing-a-New-BLE-Proxy-Tool-wp.pdf

[42] Y. Zhang, J. Weng, R. Dey, Y. Jin, Z. Lin, and X. Fu, "Breaking secure pairing of Bluetooth Low Energy using downgrade attacks," in *Proceedings of the 29th USENIX Security Symposium*, 2020, pp. 37–54. [Online]. Available: https://www.usenix.org/conference/usenixsecurity20/presentation/zhang-yue

[43] T. Sacchetti and D. Antonioli, "BLERP: BLE re-pairing attacks and defenses," in *Proceedings of the Network and Distributed System Security Symposium (NDSS)*, 2026.

[44] M. Cäsar, T. Pawelke, J. Steffan, and G. Terhorst, "A survey on Bluetooth Low Energy security and privacy," *Computer Networks*, vol. 205, p. 108712, 2022.

[45] A. Sharma, S. Sharma, S. R. Tanksalkar, S. Torres-Arias, and A. Machiry, "Rust for embedded systems: Current state and open problems," in *Proceedings of the 2024 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2024, pp. 2296–2310.

[46] Open Source Security Foundation, "SLSA: Supply-chain levels for software artifacts, version 1.0," https://slsa.dev/spec/v1.0/, 2023, accessed: 27 August 2026.

[47] in-toto Community, "in-toto: A framework to secure the integrity of software supply chains," https://in-toto.io, 2025, CNCF graduated project (graduated 23 April 2025). Accessed: 27 August 2026.

[48] M. T. Truong, N. Gruschka, and L. Lo Iacono, "You can't touch this: Detecting typosquatting packages for enhanced malware prevention in software supply chains," in *Network and System Security (NSS 2024)*, ser. Lecture Notes in Computer Science, vol. 15564. Singapore: Springer, 2025, pp. 147–166.

[49] L. Meng, X. Jiang, and D. Wu, "Adversarial robustness benchmark for EEG-based brain–computer interfaces," *Future Generation Computer Systems*, vol. 143, pp. 231–247, 2023.

[50] X. Chen, C. Liu, B. Li, K. Lu, and D. Song, "Targeted backdoor attacks on deep learning systems using data poisoning," *arXiv preprint arXiv:1712.05526*, 2017.

[51] Y. Liu, S. Ma, Y. Aafer, W.-C. Lee, J. Zhai, W. Wang, and X. Zhang, "Trojaning attack on neural networks," in *Proceedings of the 25th Annual Network and Distributed System Security Symposium (NDSS)*, 2018.

[52] L. Meng, X. Jiang, J. Huang, Z. Zeng, S. Yu, T.-P. Jung, C.-T. Lin, R. Chavarriaga, and D. Wu, "EEG-based brain–computer interfaces are vulnerable to backdoor attacks," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 31, pp. 2224–2234, 2023.

[53] L. Meng, X. Jiang, X. Chen, W. Liu, H. Luo, and D. Wu, "Adversarial filtering based evasion and backdoor attacks to EEG-based brain–computer interfaces," *Information Fusion*, vol. 107, p. 102316, 2024.

[54] M. R. Khandaker, Y. Cheng, Z. Wang, and T. Wei, "COIN attacks: On insecurity of enclave untrusted interfaces in SGX," in *Proceedings of the Twenty-Fifth International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, 2020, pp. 971–985.

[55] M.-I. Nicolae, M. Sinn, M. N. Tran, B. Buesser, A. Rawat, M. Wistuba, V. Zantedeschi, N. Baracaldo, B. Chen, H. Ludwig, I. M. Molloy, and B. Edwards, "Adversarial robustness toolbox v1.0.0," *arXiv preprint arXiv:1807.01069*, 2018.

[56] M. Tangermann, K.-R. Müller, A. Aertsen, N. Birbaumer, C. Braun, C. Brunner, R. Leeb, C. Mehring, K. J. Miller, G. Müller-Putz *et al.*, "Review of the BCI competition IV," *Frontiers in Neuroscience*, vol. 6, p. 55, 2012.

[57] I. W. Tscherniak, N. C. Thiemann, A. McWhinnie-Fernández *et al.*, "Improving motor imagery decoding methods for an EEG-based mobile brain-computer interface in the context of the 2024 Cybathlon," *Journal of NeuroEngineering and Rehabilitation*, vol. 23, no. 1, p. 129, 2026, preprint: arXiv:2511.23384.

[58] CYBATHLON ETH Zürich. (2024) Brain-computer interface (BCI) race. Archived at https://web.archive.org/web/20260124055200/https://cybathlon.com/en/event/disciplines/bci. Accessed: 27 August 2026. [Online]. Available: https://cybathlon.com/en/event/disciplines/bci

[59] X. Chen, Z. Wang, and D. Wu, "Alignment-based adversarial training (ABAT) for improving the robustness and accuracy of EEG-based BCIs," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 32, pp. 1703–1714, 2024.

[60] C. Dong, Z. Li, L. Zheng, W. Chen, and W. E. Zhang, "Boosting certificate robustness for time series classification with efficient self-ensemble," in *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management (CIKM)*, 2024, pp. 477–486.

[61] L. Li, T. Xie, and B. Li, "SoK: Certified robustness for deep neural networks," *arXiv preprint arXiv:2009.04131*, 2020.

[62] Z. Tarkhani and A. Madhavapeddy, "Enclave-aware compartmentalization and secure sharing with sirius," *arXiv preprint arXiv:2009.01869*, 2020.

[63] ——, "Enabling lightweight privilege separation in applications with MicroGuards," in *Applied Cryptography and Network Security Workshops (ACNS 2023 Workshops)*, ser. Lecture Notes in Computer Science, vol. 13907. Springer, 2023, pp. 571–598.

[64] ——, "Information flow tracking for heterogeneous compartmentalized software," in *Proceedings of the 26th International Symposium on Research in Attacks, Intrusions and Defenses (RAID)*, 2023, pp. 564–579.

[65] R. N. M. Watson, J. Woodruff, P. G. Neumann, S. W. Moore, J. Anderson, D. Chisnall, N. Dave, B. Davis, K. Gudka, B. Laurie, S. J. Murdoch, R. Norton, M. Roe, S. Son, and M. Vadera, "CHERI: A hybrid capability-system architecture for scalable software compartmentalization," in *2015 IEEE Symposium on Security and Privacy (SP)*, 2015, pp. 20–37.

[66] Z. Tarkhani, "Secure programming with dispersed compartments," Ph.D. dissertation, University of Cambridge, 2022.

[67] Sigstore Community, "Sigstore: Keyless signing for the software supply chain," https://www.sigstore.dev, 2024, accessed: 27 August 2026.

[68] ARM Limited, "ARM security technology: Building a secure system using TrustZone technology," ARM Limited, Tech. Rep. PRD29-GENC-009492C, 2009. [Online]. Available: https://documentation-service.arm.com/static/5f212796500e883ab8e74531

[69] H. Huang, F. Zhang, S. Yan, T. Wei, and Z. He, "SoK: A comparison study of Arm TrustZone and CCA," in *2024 International Symposium on Secure and Private Execution Environment Design (SEED)*, 2024, pp. 107–118.

[70] Q. Li, D. Ding, and M. Conti, "Brain-computer interface applications: Security and privacy challenges," in *Proceedings of the IEEE Conference on Communications and Network Security (CNS)*, 2015, pp. 663–666.

[71] M. Kapitonova, P. Kellmeyer, S. Vogt, and T. Ball, "A framework for preserving privacy and cybersecurity in brain-computer interfacing applications," *arXiv preprint arXiv:2209.09653*, 2022.

[72] D. Angelakis, E. Ventouras, S. Kostopoulos, and P. Asvestas, "Cybersecurity issues in brain-computer interfaces: Analysis of existing Bluetooth vulnerabilities," *Digital Technologies Research and Applications*, vol. 3, no. 2, pp. 115–139, 2024.

[73] B. Wang, Y. Yao, S. Shan, H. Li, B. Viswanath, H. Zheng, and B. Y. Zhao, "Neural cleanse: Identifying and mitigating backdoor attacks in neural networks," in *Proceedings of the 2019 IEEE Symposium on Security and Privacy (S&P)*, 2019, pp. 707–723.

[74] Y. Gao, C. Xu, D. Wang, S. Chen, D. C. Ranasinghe, and S. Nepal, "STRIP: A defence against trojan attacks on deep neural networks,"

in *Proceedings of the 35th Annual Computer Security Applications Conference (ACSAC)*, 2019, pp. 113–125.
[75] B. Chen, W. Carvalho, N. Baracaldo, H. Ludwig, B. Edwards, T. Lee, I. Molloy, and B. Srivastava, "Detecting backdoor attacks on deep neural networks by activation clustering," *arXiv preprint arXiv:1811.03728*, 2018.
[76] B. Tran, J. Li, and A. Madry, "Spectral signatures in backdoor attacks," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 31, 2018, pp. 8011–8021.
[77] J. Hayase, W. Kong, R. Somani, and S. Oh, "SPECTRE: Defending against backdoor attacks using robust statistics," in *Proceedings of the 38th International Conference on Machine Learning (ICML)*, ser. Proceedings of Machine Learning Research, vol. 139, 2021, pp. 4129–4139. [Online]. Available: https://proceedings.mlr.press/v139/hayase21a.html
[78] F. Li, M. Huang, W. You, L. Zhu, H. Cheng, and R. Yang, "Spatialspectral-backdoor: Realizing backdoor attack for deep neural networks in brain–computer interface via EEG characteristics," *Neurocomputing*, vol. 616, p. 128902, 2025.
[79] K. Liu, B. Dolan-Gavitt, and S. Garg, "Fine-pruning: Defending against backdooring attacks on deep neural networks," in *Proceedings of the 21st International Symposium on Research in Attacks, Intrusions and Defenses (RAID)*, 2018, pp. 273–294.
[80] Y. Li, X. Lyu, N. Koren, L. Lyu, B. Li, and X. Ma, "Neural attention distillation: Erasing backdoor triggers from deep neural networks," in *Proceedings of the 9th International Conference on Learning Representations (ICLR)*, 2021. [Online]. Available: https://openreview.net/forum?id=9l0K4OM-oXE
[81] D. Wu and Y. Wang, "Adversarial neuron pruning purifies backdoored deep models," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 34, 2021, pp. 16 913–16 925. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2021/hash/8cbe9ce23f42628c98f80fa0fac8b19a-Abstract.html
[82] Y. Zeng, S. Chen, W. Park, Z. M. Mao, M. Jin, and R. Jia, "Adversarial unlearning of backdoors via implicit hypergradient," in *Proceedings of the 10th International Conference on Learning Representations (ICLR)*, 2022. [Online]. Available: https://openreview.net/forum?id=MeeQkFYVbzW
[83] M. Zhu, S. Liang, and B. Wu, "Breaking the false sense of security in backdoor defense through re-activation attack," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 37, 2024, pp. 114 928–114 964. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2024/hash/d06537b4b38ccf008a54559d2c56fa23-Abstract-Conference.html
[84] A. L. Goldberger, L. A. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, and H. E. Stanley, "Physiobank, physiotoolkit, and physionet: Components of a new research resource for complex physiologic signals," *Circulation*, vol. 101, no. 23, pp. e215–e220, 2000.

APPENDIX A
ARCHITECTURES AND DATASETS

A. Datasets

- **PhysioNet Motor Movement/Imagery (MMI)** [84]. 64-channel EEG at 160 Hz; we train the baseline on S001–S010 and reserve S011–S030 for transfer-learning and backdoor studies. Preprocessing applies a 1–40 Hz band-pass, segments non-overlapping 3 s epochs (480 samples/channel), standardizes each channel to zero mean/unit variance, and uses a 70%/15%/15% train/-val/test split.
- **Sleep-EDF**. For sleep staging with DeepSleepNet, EEG is segmented into 30 s windows, normalized as in the original pipeline, and evaluated following the protocol in [29].
- **BCI Competition IV 2a**. BCI Competition IV 2a [56]. This dataset consists of EEG recordings from nine healthy subjects performing motor imagery of four classes: left hand, right hand, both feet, and tongue movements. The signals were recorded using 22 EEG channels at 250 Hz while the participants followed on-screen cues during multiple sessions. Each session contains 288 trials (72 per class), organised as six runs of 48 trials, with 4-s imagery periods. This dataset is widely used as a benchmark for evaluating EEG MI classification models. In this work, only the three sensorimotor channels (C3, Cz, C4) that correspond to the left, central, and right motor cortices were used. These channels were resampled to 128 Hz to be compatible with the replay experiments.
- **NeuroTUM** [57]. EEG recordings collected using a 24-channel Smarting headset during a motor imagery paradigm developed for the CYBATHLON 2024 competition. Rest epochs were extracted from "CIRCLE BLACKSCREEN" event markers and mapped to the 14 channels overlapping with the BCI-IV-2a montage.

B. Architectures

- **EEGNet** [28]. Inputs are 64×480 (channels$\times$time). The network applies a temporal convolutional block (frequency-specific structure), a depthwise spatial convolution (channel-wise filtering), and a separable convolution (joint temporal–spatial features), with ELU activations, batch normalization, and average pooling, followed by a fully connected head with max-norm and softmax. Training uses Adam with categorical cross-entropy; on the held-out MMI split the model reaches **100%** accuracy with per-class precision/recall/F1 = 1.00.
  For the replay-attack experiments we used EEGNetv4 as implemented in Braindecode. The model is a standard EEGNetv4 architecture pre-trained on the BCI Competition IV 2a (BNCI2014001) data using only 3-classes of motor imagery (feet, left hand, right hand). The network expects 3 channels (C3, Cz, C4) sampled at 128 Hz with an input window length of 3.01 s (385 samples). No further fine-tuning was done for the replay experiments.
- **DeepSleepNet** [29]. We use the official TensorFlow implementation[1] unchanged. Training follows the original two-stage regime (pretrain convolutional feature extractor, then fine-tune the full network) on Sleep-EDF, and evaluation on a held-out set yields $\sim$86% accuracy (macro-F1 computed per the original protocol).

APPENDIX B
EEGLE FRAMEWORK: FEEDBACK ENGINE ABLATION

To validate that the AI-Assisted Feedback Engine Φ provides coverage beyond static sequential engine execution, we compare two configurations: *Static-EEGle*, which runs the three foundational engines once in a fixed order (Evasion $\rightarrow$ Backdoor $\rightarrow$ System) without LLM-driven replanning; and *EEGle*+Φ, which uses the full feedback loop.

Table IX shows that Static-EEGle misses the E-desynchronisation (TDA) dimension entirely—the static

[1] https://github.com/akaraspt/deepsleepnet

TABLE IX: Ablation of the AI-Assisted Feedback Engine Φ. "Coverage" = fraction of V1–V4 sub-surfaces and NERVE dimensions for which at least one successful attack instance was confirmed. "Novel engines" = engines synthesised at runtime by Φ's generative extension that were not in the initial repertoire.

Config.	NERVE dims. covered	V sub-surfaces confirmed	Iterations	Novel engines
Static-EEGle	4/5	3/4	3	0
EEGle+Φ	5/5	4/4	11	3

evasion engine applies only standard L_∞-bounded perturbations and does not attempt temporal injection without explicit direction from Φ. It also fails to confirm V4 (memory-unsafe SDK), because exploiting the confused-deputy path requires cross-referencing the System engine's format-string findings with the Evasion engine's injection capability—a cross-dimension inference that only Φ's stateful analysis performs. The three LLM-synthesised engines generated at runtime included a BLE replay-injection module, an augmented-NRA variant targeting cross-session fingerprinting, and a supply-chain simulation engine for E1 dependency confusion. These would not exist in a static repertoire. The additional iterations (11 vs. 3) reflect Φ issuing refinement plans in response to partial failures—for example, discovering that the initial NFA parameters failed preprocessing and re-issuing with corrected spectral envelopes.

APPENDIX C
SYNTHETIC EEG DATA GENERATION

Motor Imagery. Motor imagery (MI) involves mentally rehearsing motor actions, leading to distinct EEG patterns that reflect cortical activation and inhibition dynamics. Rather than constructing a baseline signal from parametric components, we synthesise MI trials by applying physiologically motivated frequency-band suppression directly to real resting-state epochs. This preserves the spectral structure, $1/f$ characteristics, artifacts, and cross-channel covariance of genuine neural recordings while producing class-discriminative ERD patterns indistinguishable—to a preprocessing pipeline—from real MI activity.

a) Rest Epoch as Baseline.: Let $x_{\text{ref}} \in \mathbb{R}^{C\times T}$ denote a resting-state epoch with C channels and T time samples, drawn from a public dataset (for example, BCI Competition IV 2a [56] or neuroTUM [57]). This epoch serves as the baseline signal directly:

$$\text{EEG}_{\text{baseline}}(ch,t) = x_{\text{ref}}(ch,t) \tag{5}$$

b) Motor Imagery Modulation.: MI patterns are realised by subtracting band-limited components of the rest epoch at class-specific spatial locations, mimicking event-related desynchronization (ERD):

$$\hat{x} = x_{\text{ref}} - S_\mu - S_\beta \tag{6}$$

where each suppression term is defined as:

$$S_\mu = D_\mu \cdot \text{BP}_{[8,12]}(x_{\text{ref}}) \odot M(c^*) \tag{7}$$

$$S_\beta = D_\beta \cdot \text{BP}_{[13,30]}(x_{\text{ref}}) \odot M(c^*) \tag{8}$$

Here $\text{BP}_{[f_1,f_2]}$ denotes 4th-order zero-phase Butterworth band-pass filtering (implemented via `sosfiltfilt`), $D_\mu = 0.90$ and $D_\beta = 0.70$ are the suppression depths for the mu and beta bands respectively, $M(c^*)$ is a class-specific spatial mask, and $\odot$ denotes element-wise multiplication.

c) Spatial Mask.: Suppression is centred on the focal electrode for the target class and falls off as a Gaussian over 2D Euclidean scalp distance:

$$M(ch) = \exp\left(-\frac{d(ch,\,\text{focal}(c^*))^2}{2\sigma^2}\right), \quad \sigma = 0.40 \tag{9}$$

The focal electrode assignments follow established MI neurophysiology:

Target class	Focal channel	Physiological basis
Left hand MI	C4	Contralateral (right hemisphere) ERD
Right hand MI	C3	Contralateral (left hemisphere) ERD
Feet MI	Cz	Central midline ERD

d) Temporal Envelope.: A sigmoid function ramps suppression in over time:

$$E(t) = \frac{1}{1+\exp\big(-k\cdot(t-t_{\text{onset}})\big)}, \quad k = 15.0,\ t_{\text{onset}} = 0.0\ \text{s} \tag{10}$$

e) Complete Combination.: The complete synthetic signal is thus:

$$\hat{x} = x_{\text{ref}} - D_\mu\cdot\text{BP}_{[8,12]}(x_{\text{ref}})\odot M(c^*) - D_\beta\cdot\text{BP}_{[13,30]}(x_{\text{ref}})\odot M(c^*) \tag{11}$$

No additional noise or artifact injection is required: all physiological noise, blink artifacts, and non-stationarity present in x_{ref} are retained in $\hat{x}$, which is precisely what makes the synthetic signal plausible to the preprocessing pipeline. Producing a different MI class from the same rest epoch requires only changing the focal electrode in $M(c^*)$, so arbitrarily many labelled trials can be generated on demand from a single reference recording.

APPENDIX D
NRA PIPELINE DETAILS

Augmentation in the frequency space starts by transforming each EEG channel (or each epoch–channel pair) into the frequency domain via a Fast Fourier Transform (FFT). For a given channel, the complex spectrum was expressed in terms of amplitude and phase components, $A(f)$ and $\phi(f)$, respectively. To introduce controlled variability, small relative Gaussian noise was applied to the amplitudes:

$$A'(f) = A(f)\cdot(1+\varepsilon), \quad \varepsilon \sim \mathcal{N}(0,\sigma)$$

where σ denotes a small standard deviation (e.g., 0.01–0.05). Optionally, the noise magnitude can be modulated across frequency bands—for example, using a smaller σ_{in} within

discriminative ranges such as the mu or beta bands (e.g., $\sigma_{\text{in}} = 0.005$, $\sigma_{\text{out}} = 0.03$).

The perturbed signal was then reconstructed in the time domain using the inverse FFT (IFFT):

$$x'(t) = \text{IFFT}\left(A'(f) \cdot e^{i\phi(f)}\right)$$

The implemented function operates on an EEG epoch of shape $(3, 385)$, corresponding to three channels with 385 samples each. It performs the FFT per channel, perturbs each frequency bin's amplitude according to the formulation above, and reconstructs the modified signal. Optionally, specific frequency bands can be preserved by applying reduced noise within those regions.

To introduce minimal temporal variability while preserving the overall oscillatory structure, a small random perturbation was applied to the phase spectrum. Specifically, the phase of each frequency component was jittered according to

$$\phi'(f) = \phi(f) + \delta, \quad \delta \sim \mathcal{N}(0, \sigma_{\text{phase}})$$

We add Gaussian-distributed phase noise, centered at zero, independently to each frequency bin. The standard deviation, σ_{phase}, is kept small (e.g., 0.01–0.1 radians) to prevent temporal distortion. This method is considered safer for oscillatory motor imagery paradigms where class information is mainly in the mu and beta band amplitudes, not precise phase.

To introduce controlled spectral variability, the amplitude spectrum is modulated in specific frequency bands $f \in [f_{\text{low}}, f_{\text{high}}]$. The amplitude is scaled by a small, uniformly distributed random factor $k \sim \mathcal{U}(-\alpha, \alpha)$, such that $A'(f) = A(f) \cdot (1 + k)$. For example, the 8–13 Hz mu band can be multiplied by $1 + k$ to introduce mild, shape-preserving variability ($\alpha = 0.03$). This function allows users to specify arbitrary ranges and scaling intervals. Unlike global noise addition, this selective modulation targets physiologically relevant bands, making it a more interpretable and neurophysiologically grounded augmentation strategy.

A conservative composite augmentation strategy was also implemented, mildly perturbing both amplitude and phase concurrently. This "combination" approach makes small adjustments to spectral magnitude and phase while preserving critical discriminative frequency bands (e.g., μ and β rhythms) for MI classification. The modified signal is formally:

$$x'(t) = \text{IFFT}\left(A'(f) \cdot e^{i\phi'(f)}\right),$$

where $A'(f)$ and $\phi'(f)$ use the respective noise models with reduced perturbation magnitudes compared to independent application. This results in conservative spectral augmentation, maintaining physiological plausibility and enhancing data diversity.

APPENDIX E
WHITE-BOX ATTACKS: FORMAL DEFINITIONS AND ALGORITHMIC DETAILS

A. Notation

Let $\mathbf{X} \in \mathbb{R}^d$ denote an input, y its true label, $f(\cdot)$ the model logits (or predictive function) and $L(\mathbf{X}, y)$ the scalar loss used for training (e.g., cross-entropy). Let $\mathcal{T}$ denote a distribution of preprocessing transforms for EOT; $\mathbb{E}_{\tau\sim\mathcal{T}}[\cdot]$ denotes expectation over τ. For a perturbation budget ϵ we write $B_\epsilon(\mathbf{X})$ for the corresponding norm ball (e.g., L_∞ or L_2).

B. Fast Gradient Sign Method (FGSM)

A single-step L_∞ update for an untargeted attack is given by

$$\mathbf{X}^{\text{adv}} = \mathbf{X} + \epsilon \, \text{sign}\big(\nabla_{\mathbf{X}} L(\mathbf{X}, y)\big). \tag{12}$$

To reduce label leaking when the true label is not used, replace y with the model prediction $\hat{y} = \arg\max f(\mathbf{X})$. For targeted FGSM, use $-\nabla_{\mathbf{X}} L(\mathbf{X}, y^\star)$ where $y^\star$ is the target class.

C. Projected Gradient Descent (PGD)

We run T iterations with step size α and project each iterate back into $B_\epsilon(\mathbf{X})$. The attack is initialized from the clean input because random initialization was disabled:

$$g_t := \nabla_{\mathbf{X}} L\big(f(\mathbf{X}_t^{\text{adv}}),\, y\big), \tag{13}$$

$$\mathbf{X}_0^{\text{adv}} = \mathbf{X}, \tag{14}$$

$$\mathbf{X}_{t+1}^{\text{adv}} = \Pi_{B_\epsilon(\mathbf{X})}\big(\mathbf{X}_t^{\text{adv}} + \alpha \, \text{sign}(g_t)\big). \tag{15}$$

Here, $\Pi_{B_\epsilon(\mathbf{X})}(\cdot)$ denotes projection onto the feasible perturbation ball. For L_2 variants, the sign update is replaced by a normalized gradient direction. We stop early when the example becomes misclassified.

D. Carlini & Wagner (C&W)

C&W formulates an unconstrained optimization that trades perturbation norm against an attack loss. For the L_2 formulation used here:

$$\min_{\delta} \; \|\delta\|_2^2 \; + \; c \, \mathbb{E}_{\tau\sim\mathcal{T}}\big[g\big(f(\tau(\mathbf{X} + \delta)), y\big)\big], \tag{16}$$

where $g(\cdot)$ is a differentiable misclassification objective (e.g., margin-based) and $c > 0$ is a weighting constant. Optimization proceeds with Adam and optional restarts over c to identify a minimal perturbation achieving success. The perturbation is constrained implicitly by a penalty or via an explicit projection step, depending on the chosen implementation.

E. DeepFool

Assuming a differentiable classifier with decision boundaries locally approximated by linear functions, DeepFool iteratively computes minimal perturbations to cross the closest boundary. At iteration i:

$$\delta_i = -\frac{f(\mathbf{X}_i)}{\|\nabla f(\mathbf{X}_i)\|_2^2} \nabla f(\mathbf{X}_i), \tag{17}$$

$$\mathbf{X}_{i+1} = \mathbf{X}_i + \delta_i, \tag{18}$$

and the procedure repeats until the predicted label changes. The total perturbation is $\delta = \sum_i \delta_i$, which serves as an estimate of the minimal L_2 perturbation required.

F. Implementation notes

- All attacks were executed using the same preprocessing pipeline to ensure fair comparison. EOT was not enabled for the white-box robustness probe.
- For iterative attacks, we used early stopping on misclassification and report perturbation norms measured on the final clipped example.
- Attack budgets and iteration counts are given in the caption of Table VI. PGD used a step size of $\alpha = 0.1$ (`eps_step` in ART). For C&W, the initial optimization constant was $c = 0.01$ (`initial_const`), with `binary_search_steps`=10. No explicit random seed was set, and PGD random initialization was disabled.

APPENDIX F
CROSS-LAYER ATTACK CHAIN: V→N END-TO-END

The V and N dimensions are not independent: Vein Tapping provides the injection channel through which NFA payloads reach the target model. To demonstrate this cross-layer dependency explicitly, we performed an end-to-end V→N chain on NeuroSky MindWave Mobile 2.

Setup. Using the V1 passive-sniffing configuration (nRF52840 dongle, Wireshark GATT plugin), we first established the BLE advertisement timing and GATT handle layout of the target device. We then deployed the V2 MitM configuration (two Raspberry Pi 4 nodes running GATTacker), establishing a transparent relay between the headset and the host application. From this relay position we can observe, modify, or replace any epoch before it reaches the BrainFlow ingestion layer.

Injection. NFA payloads (A1–A3 from Table I) were pregenerated offline and serialised to the raw byte layout expected by the NeuroSky TGAM packet format (24-bit attention/meditation values with checksum). The relay intercepts the real EEG packet, discards it, and injects the NFA-synthesised payload within the same BLE connection interval ($\leq$7.5 ms), making the substitution transparent to the host.

Result. The BrainFlow mental-state classifier received the injected epochs and produced the attacker-chosen class output in 100 % of injection attempts across 20 trials, with a mean end-to-end latency from interception to misclassification of 9.3 ms (dominated by BrainFlow's preprocessing pipeline). The host application observed no anomaly: packet timing, checksum, and attention-value range were all within normal bounds. This confirms that the V–N attack chain is not a theoretical composition but an empirically executable exploit requiring only commodity hardware and the NFA payload generation capability provided by EEGle.

APPENDIX G
EXPERIMENTAL SETUP

Table X lists the workstation, software, and BCI hardware used for every experiment reported in Section VII. The mapping from each platform to the NERVE dimension it was used to evaluate is given in Table III.

TABLE X: System Configuration

Category	**Details**
Workstation	Alienware Aurora R16
Operating System	Ubuntu 25.04 (Linux kernel 6.14.0-34)
Hardware	NVIDIA GeForce RTX 4070 Ti GPU; 64 GB RAM
Programming Environment	Python 3.13.3
Core Libraries	TensorFlow 2.20.0; NumPy 2.3.2; SciPy 1.16.1; scikit-learn 1.7.1; Pandas 2.3.2; Matplotlib 3.10.6; MNE 1.10.1
BCI Tools	BrainFlow Library 5.18.0
Devices	OpenBCI Cyton (8-channel, 32-bit); NeuroSky MindWave Mobile 2; Muse2

APPENDIX H
MODEL TRAINING HYPERPARAMETERS

The hyperparameters for the EEGNet base model and for the transfer learning setup used in the backdoored multi-trigger model are summarised in Figure 6 and Figure 7, respectively.

APPENDIX I
BACKDOOR FIGURES

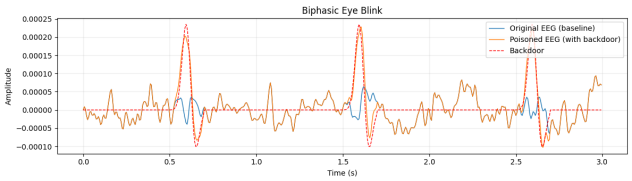
Figure 8 summarises the ten backdoor waveform templates we consider, illustrating how each pattern is injected into clean EEG to produce the poisoned signals used in our experiments.

Category	Hyperparameter	Value
Data Sampling & Segmentation	Sampling Rate	160 Hz
	Epoch Length	3 seconds
	Segment Length	480 samples
Data Split	Train / Val / Test	70% / 15% / 15%
Preprocessing	Band-pass Filter	1–40 Hz (IIR)
	Normalization	StandardScaler (per-segment, channel-wise)
Model Architecture (EEGNet)	Channels	64
	Samples per Epoch	480
	Temporal Kernel Length	64
	F_1 (Temporal Filters)	8
	Depth Multiplier (D)	2
	F_2 (Separable Filters)	16
	Dropout Rate	0.5
	Activation Function	ELU
Training Parameters	Optimizer	Adam
	Learning Rate	0.001
	Loss Function	Categorical Cross-Entropy
	Batch Size	32
	Epochs	100 (early stopping)
Callbacks	EarlyStopping	Patience 15, monitor: val_accuracy
	Reduce LR on Plateau	Factor 0.2, Patience 3, Min LR $= 10^{-7}$
	Model Checkpoint	Monitor: val_accuracy

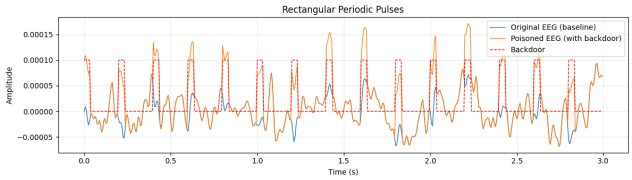
Fig. 6: EEGNet Base Model Hyperparameters

Category	Hyperparameter	Value
Data Sampling & Segmentation	Sampling Rate	160 Hz
	Epoch Length	3 seconds
Data Split	Train / Val / Test	70% / 15% / 15%
Preprocessing	Band-pass Filter	1–40 Hz (IIR)
	Normalization	None (TL stage)
Backdoor Injection	Injection Strategy	Added to all epochs of class
	Channel Application	All channels of poisoned samples
Transfer Learning Setup	Trainable Layers	All unfrozen
Training Parameters	Optimizer	Adam
	Learning Rate	0.0005
	Loss Function	Categorical Cross-Entropy
	Batch Size	32
	Epochs	100 (early stopping)
Callbacks	EarlyStopping	Patience 10, monitor: val_accuracy
	Reduce LR on Plateau	Factor 0.2, Patience 5, Min LR $= 10^{-7}$

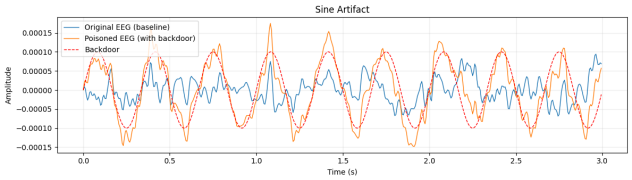
Fig. 7: Transfer Learning Hyperparameters for Backdoored Multi-Trigger Model



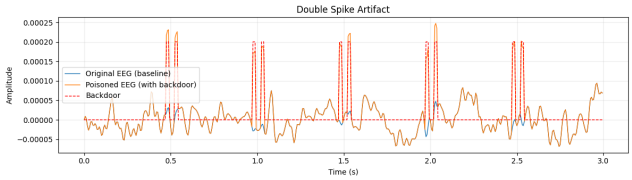
(a) Biphasic Eye Blink - Top: original EEG; Middle: poisoned EEG with injected blink template; Bottom: Biphasic eye-blink backdoor pattern.



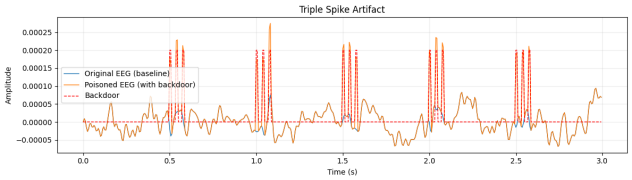
(b) Rectangular Periodic Pulse - clean vs. poisoned EEG and backdoor waveform.



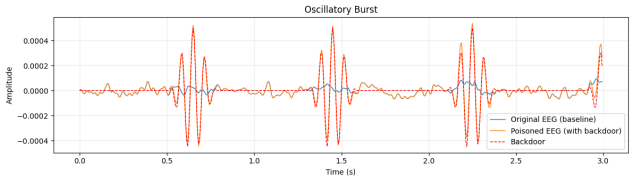
(c) Continuous Sine - original, poisoned, and backdoor waveform (3 Hz sine).



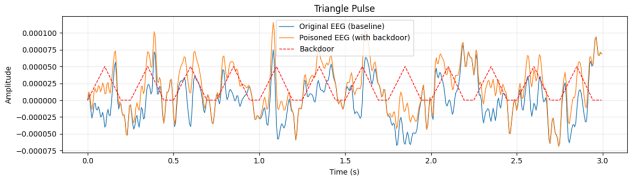
(d) Double Spike - original vs. poisoned EEG and injected double-spike backdoor.



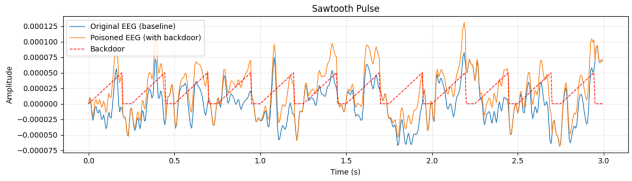
(e) Triple Spike - clean vs. poisoned EEG and backdoor waveform.



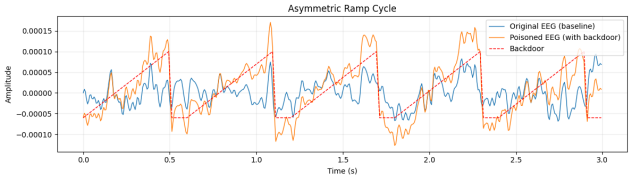
(f) Oscillatory Burst - original, poisoned, and backdoor waveform (15 Hz rhythmic bursts).



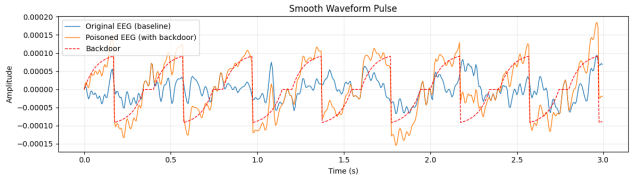
(g) Triangle Pulse - original vs. poisoned EEG and backdoor triangular pulses.



(h) Sawtooth Pulse - original vs. poisoned EEG and injected sawtooth waveform.



(i) Asymmetric Ramp Cycle - original vs. poisoned EEG and injected ramp waveform.



(j) Smooth Waveform Pulse - original vs. poisoned EEG and injected smooth waveform pattern.

Fig. 8: Overview of backdoor waveform templates injected into EEG signals across multiple patterns.