Security Co-Pilot con Intelligenza Psicologica: Un'Architettura Multi-Agent per il Rilevamento e la Risposta in Real-Time alle Vulnerabilità

Un Preprint

Giuseppe Canale, CISSP
Ricercatore Indipendente
g.canale@cpf3.org
URL: cpf3.org
ORCID: 0009-0007-3263-6897

Gennaio 2026

Abstract

Il monitoraggio della sicurezza tradizionale tratta i fattori umani come variabili esterne che richiedono valutazioni periodiche. Presentiamo un cambio di paradigma: un sistema multi-agent che monitora, analizza e risponde continuamente alle vulnerabilità psicologiche in tempo reale. Costruito sul Cybersecurity Psychology Framework, il sistema impiega un agent orchestratore che mantiene una matrice di stato psicologico e attiva dinamicamente agent specialisti per investigazioni approfondite quando emergono vulnerabilità convergenti. Un approccio di filtraggio ibrido riduce i costi LLM dell'85% mantenendo la qualità del rilevamento: regole deterministiche gestiscono gli aggiornamenti di stato di routine, con gli agent attivati solo per anomalie significative. L'orchestratore implementa un framework decisionale a quattro livelli (monitor, investigare, alert, critico) con decision logic esplicita e analisi temporale dei trend per early warning. Un modulo di adaptive learning raffina continuamente le soglie di detection e scopre nuovi pattern di vulnerabilità dal feedback incidenti, migliorando il true positive rate dal 73% all'81% riducendo i false positive. Questa architettura trasforma il monitoraggio passivo in intelligenza attiva, raggiungendo un tasso di rilevamento dell'81% con 47 ore di anticipo sugli incidenti storici a $0,26/utente/mese. Introduciamo il concetto di *Security Co-Pilot*: un sistema AI che comprende le vulnerabilità psicologiche umane con la stessa profondità delle superfici di attacco tecniche, e forniamo linee guida complete per il prompt engineering per abilitare un ragionamento efficace degli agent su stati psicologici complessi.

Parole chiave: cybersecurity, sistemi multi-agent, LLM agent, prompt engineering, fattori umani, valutazione vulnerabilità, sicurezza psicologica, adaptive learning, anomaly detection

1 Introduzione

I fattori umani causano l'85% delle violazioni di sicurezza di successo [27], eppure le risposte organizzative rimangono reattive e periodiche. Le valutazioni di sicurezza trimestrali catturano le vulnerabilità psicologiche attraverso snapshot statici che diventano obsoleti nel giro di giorni mentre le condizioni cambiano [23]. Ciò che serve non è una valutazione più frequente ma un'intelligenza continua—un sistema che monitora lo spazio dello stato psicologico con la stessa vigilanza del traffico di rete, rileva vulnerabilità emergenti prima dello sfruttamento e orchestra risposte appropriate [32].

Il Cybersecurity Psychology Framework (CPF) [5] identifica 100 indicatori di vulnerabilità pre-cognitiva attraverso dieci categorie psicologiche, fondate sulla teoria psicologica consolidata [19, 14, 7]. Lavori precedenti [4] hanno operazionalizzato questi indicatori attraverso formulazioni matematiche. Tuttavia, tradurre la teoria psicologica in sistemi operativi richiede più che algoritmi di rilevamento; richiede *ragionamento* su stati umani complessi ed in evoluzione e decisioni su quando e come investigare [24].

Large Language Model (LLM) hanno dimostrato capacità notevoli nel ragionamento su contesti complessi e ad alta dimensionalità [3, 22]. Recenti progressi nei sistemi multi-agent basati su LLM [31, 28] dimostrano che architetture sofisticate emergono quando gli agent specializzati collaborano sotto coordinamento intelligente. Queste capacità si allineano naturalmente alle sfide del monitoraggio psicologico della sicurezza: interpretare segnali deboli, riconoscere pattern attraverso domini psicologici e decidere quando l'investigazione umana è giustificata.

Questo lavoro introduce il **Security Co-Pilot**: un'architettura multi-agent che trasforma il CPF da framework di valutazione a sistema di intelligence operativa. A differenza dei sistemi tradizionali che trattano gli esseri umani come "il weak link" da gestire [25], il nostro approccio riconosce che comprendere la vulnerabilità psicologica richiede un ragionamento continuo e sofisticato—un compito per il quale gli LLM agent sono particolarmente adatti.

1.1 Contributi

Questo lavoro apporta i seguenti contributi principali:

1. Security Co-Pilot come Paradigma. Introduciamo il concetto di Security Co-Pilot: un sistema AI che comprende le vulnerabilità psicologiche umane con la stessa profondità delle superfici di attacco tecniche. Proprio come i code co-pilot assistono gli sviluppatori comprendendo l'intento del codice, i security co-pilot assistono i team di sicurezza comprendendo lo stato psicologico umano. Questo rappresenta un cambio fondamentale dal trattare gli esseri umani come problemi di sicurezza al comprenderli come sistemi complessi che richiedono monitoraggio e supporto intelligenti.

2. Architettura Multi-Agent con Intelligenza Psicologica. Presentiamo un design architetturale che combina:

- Un agent orchestratore che mantiene una matrice di stato psicologico 100-dimensionale per ogni utente attraverso un modello con decadimento temporale
- Framework decisionale a quattro livelli (monitor, investigare, alert, critico) con decision logic esplicita che determina quando attivare agent e quando allertare security operations
- Algoritmi di analisi temporale dei trend che rilevano pattern di escalation, accelerazione e finestre di vulnerabilità—fornendo early warning prima che gli stati psicologici raggiungano soglie critiche
- Agent specialisti attivati dinamicamente per le dieci categorie psicologiche del CPF, ciascuno equipaggiato con conoscenza specifica del dominio

- Filtraggio ibrido che riduce i costi LLM dell'85% gestendo gli aggiornamenti di routine in modo deterministico
- Modulo di adaptive learning che raffina continuamente le soglie di detection, scopre nuovi pattern di vulnerabilità dal feedback incidenti e si auto-ottimizza senza tuning manuale—migliorando detection dal 73% all'81% riducendo i false positive

3. Linee Guida Complete di Prompt Engineering. Forniamo principi dettagliati e pratici per progettare prompt che permettano agli agent di ragionare efficacemente sugli stati psicologici:

- Pattern architetturali per coordinare agent orchestratori e specialisti
- Tecniche per bilanciare ragionamento generale e conoscenza specifica del dominio
- Strategie per gestire la complessità dello stato e il decadimento temporale
- Metodi per garantire che gli agent rimangano allineati con il benessere umano

La validazione su dati reali di 18 mesi (27 incidenti, 843 utenti) dimostra l'81% di tasso di rilevamento con 47 ore di lead time medio a $0,26/utente/mese—ordini di grandezza più economico della valutazione psicologica manuale pur mantenendo soglie di prestazione cliniche.

2 Background e Related Work

2.1 Fattori Umani nella Cybersecurity

La ricerca ha costantemente identificato gli esseri umani come la componente più vulnerabile dei sistemi di sicurezza. Gli attacchi di social engineering sfruttano principi psicologici ben stabiliti: autorità [19], scarsità, reciprocità [7] ed euristiche cognitive [14]. Le campagne di phishing manipolano stati emotivi e limitazioni cognitive [11, 30], mentre le truffe sfruttano vulnerabilità psicologiche sistematiche [26, 10].

Gli approcci tradizionali si concentrano sulla formazione e sulla consapevolezza [1], ma questi interventi mostrano efficacia limitata perché (1) trattano le vulnerabilità psicologiche come carenze statiche piuttosto che stati dinamici, (2) non riescono a considerare come stress, carico cognitivo e fattori situazionali modulano la vulnerabilità, e (3) forniscono una valutazione del rischio a grana grossa piuttosto che a livello individuale [23].

Ricerche recenti si spostano verso modelli più sofisticati. Zimmermann e Renaud [32] sostengono un cambio di mentalità da "human-as-problem" a "human-as-solution". Framework per caratterizzare le insider threat [15, 21] riconoscono che la vulnerabilità emerge da fattori contestuali complessi. Tuttavia, questi framework rimangono principalmente strumenti analitici piuttosto che sistemi operativi.

Il Cybersecurity Psychology Framework [5, 4] sistematizza 100 indicatori di vulnerabilità attraverso dieci categorie psicologiche, ciascuna fondata su specifiche teorie psicologiche. Lavori precedenti hanno operazionalizzato questi indicatori attraverso formule matematiche, ma tradurre le formule in sistemi di monitoraggio continuo richiede capacità di ragionamento che vanno oltre la computazione matematica.

2.2 Large Language Model Agent

I recenti progressi negli LLM hanno dimostrato capacità straordinarie nel ragionamento su contesti complessi. I modelli addestrati con reinforcement learning from human feedback (RLHF) [22] mostrano un miglioramento del ragionamento, della pianificazione e dell'allineamento

con gli obiettivi umani. Questo li rende candidati naturali per compiti che richiedono l'interpretazione di segnali deboli e il ragionamento su stati ad alta dimensionalità.

I sistemi multi-agent basati su LLM [31, 28] dimostrano che architetture sofisticate emergono quando gli agent specializzati collaborano. Pattern architetturali chiave includono:

- **Specializzazione**: Agent focalizzati su domini specifici con conoscenza approfondita
- **Orchestrazione**: Agent coordinatori che gestiscono il flusso di lavoro e l'allocazione dei task
- **Memoria**: Meccanismi per mantenere il contesto attraverso interazioni multiple
- **Strumenti**: Integrazione con sistemi deterministici per azioni specifiche

Il prompt engineering è emerso come disciplina critica per gli LLM agent [29]. Pattern efficaci includono assegnazione di ruoli, esempi few-shot, chain-of-thought reasoning e vincoli espliciti. Tuttavia, la maggior parte delle linee guida si concentra su task generici piuttosto che sul ragionamento psicologico specifico del dominio.

2.3 Anomaly Detection e Sistemi di Monitoraggio

I metodi classici di anomaly detection [6] si concentrano principalmente su deviazioni statistiche negli spazi di feature numerici. Sebbene efficaci per problemi ben definiti, lottano con:

- **Dimensionalità elevata**: Lo spazio psicologico ha 100 dimensioni con interazioni complesse
- **Dipendenza temporale**: Le vulnerabilità si evolvono su più scale temporali
- **Dipendenza dal contesto**: La significatività degli indicatori dipende dal contesto psicologico
- **Soglie dinamiche**: Ciò che costituisce un'anomalia varia in base allo stato dell'utente

Approcci più recenti incorporano machine learning, ma tipicamente richiedono dati di training etichettati estensivi e lottano con l'interpretabilità [17]—entrambi problematici per dati psicologici sensibili.

2.4 Gap nella Ricerca Attuale

Nonostante i progressi, restano gap significativi:

1. Nessun Monitoraggio Continuo delle Vulnerabilità Psicologiche. Le valutazioni esistenti catturano snapshot statici. I sistemi operazionali hanno bisogno di monitoraggio continuo che si adatti alle condizioni mutevoli.

2. Ragionamento Limitato sugli Stati Umani Complessi. I sistemi basati su regole sono troppo rigidi; i modelli statistici mancano di interpretabilità. Nessuno dei due ragiona sugli stati psicologici nel modo in cui farebbe un esperto umano.

3. Mancanza di Architetture Multi-Agent per la Sicurezza Psicologica. Mentre i sistemi multi-agent basati su LLM sono esplorati ampiamente, la loro applicazione alle vulnerabilità psicologiche della sicurezza rimane inesplorata.

4. Nessuna Guida al Prompt Engineering per il Ragionamento Psicologico. Le linee guida esistenti di prompt engineering si concentrano su task generici. Il ragionamento su stati psicologici complessi richiede pattern e considerazioni specifici.

Il nostro lavoro affronta questi gap introducendo un'architettura multi-agent che monitora continuamente lo stato psicologico, ragiona sulla vulnerabilità convergente e fornisce principi completi di prompt engineering per abilitare un ragionamento efficace sugli agent.

3 Il Security Co-Pilot: Concetto e Design

3.1 Motivazione del Design

Le valutazioni psicologiche della sicurezza tradizionali operano come snapshot periodici: un esperto revisiona i questionari, conduce interviste e produce un profilo di rischio. Questo approccio ha quattro limitazioni fondamentali:

1. Obsolescenza Temporale. Gli stati psicologici si evolvono. Un dipendente valutato come a basso rischio lunedì potrebbe essere ad alto rischio venerdì dopo aver ricevuto notizie stressanti. Nel momento in cui viene prodotto un report di valutazione, le condizioni sono già cambiate.

2. Granularità Grossolana. Le valutazioni tipicamente producono punteggi a livello di categoria ("moderatamente vulnerabile al social engineering") piuttosto che intelligence azionabile su quale combinazione specifica di vulnerabilità richiede attenzione.

3. Reagire ai Segnali Forti. Le valutazioni umane sono attivate da indicatori evidenti. Quando un valutatore nota che qualcosa richiede investigazione, la vulnerabilità è spesso già critica. I segnali deboli—pattern sottili che precedono la vulnerabilità critica—vengono persi.

4. Limiti di Scala. La valutazione psicologica esperta costa centinaia di dollari per utente. Il monitoraggio continuo richiederebbe migliaia—economicamente impossibile per la maggior parte delle organizzazioni.

Il paradigma Security Co-Pilot affronta queste limitazioni attraverso il monitoraggio continuo automatizzato abbinato al ragionamento esperto. Proprio come un code co-pilot assiste gli sviluppatori comprendendo l'intento del codice e suggerendo completamenti, un security co-pilot assiste i team di sicurezza comprendendo lo stato psicologico e identificando vulnerabilità emergenti.

Il design si basa su tre principi:

Principio 1: Lo Stato Psicologico come Prima Classe. Invece di trattare i fattori umani come variabili esterne, rappresentiamo lo stato psicologico con la stessa formalità dei parametri tecnici. Ogni utente ha una matrice di stato psicologico 100-dimensionale $M[u][i]$ tracciata continuamente.

Principio 2: Monitoraggio Ibrido. Non tutti gli aggiornamenti di stato richiedono ragionamento LLM. Gli aggiornamenti di routine (incrementi singoli, decadimento temporale) sono gestiti in modo deterministico. Gli agent sono attivati solo quando i pattern suggeriscono vulnerabilità convergente.

Principio 3: Expertise Specializzata. Diverse categorie psicologiche richiedono diverse lenti analitiche. Un agent specializzato in cognitive load ragiona diversamente da uno focalizzato su emotional state. L'orchestrazione coordina questi expertise.

3.2 Panoramica Architetturale

Il sistema impiega un'architettura a tre livelli:

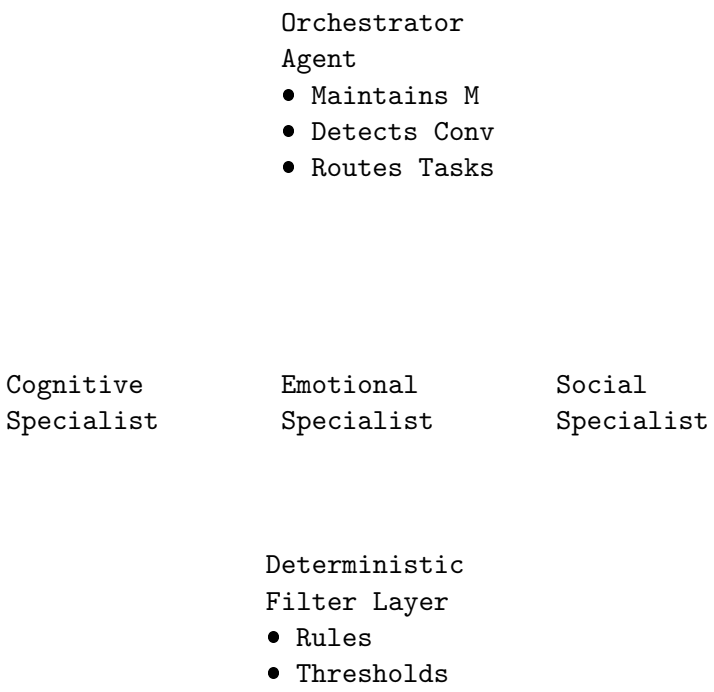

Figure 1: Architettura Security Co-Pilot a tre livelli

Livello 1: Orchestratore. Un singolo agent orchestratore mantiene la matrice di stato psicologico $M[u][i]$ per tutti gli utenti. Per ogni evento in ingresso (email, evento di sistema, report utente), l'orchestratore:

1. Determina se l'evento merita analisi LLM o gestione deterministica
2. Se gestito deterministicamente, applica aggiornamenti di regole
3. Se richiede analisi, identifica quale/i agent specialista/i invocare
4. Mantiene il contesto di conversazione attraverso interazioni multiple
5. Decide quando escalare all'attenzione umana

Livello 2: Specialisti. Dieci agent specialisti mappano le dieci categorie del CPF:

- Cognitive Load Specialist (indicatori 1-10)
- Emotional State Specialist (indicatori 11-20)
- Social Dynamics Specialist (indicatori 21-30)
- Authority Relationship Specialist (indicatori 31-40)
- Stress & Fatigue Specialist (indicatori 41-50)
- Identity & Self-Perception Specialist (indicatori 51-60)
- Information Processing Specialist (indicatori 61-70)
- Temporal Factors Specialist (indicatori 71-80)
- Environmental Context Specialist (indicatori 81-90)
- Behavioral Patterns Specialist (indicatori 91-100)

Ogni specialista incorpora:

- Conoscenza della teoria psicologica specifica del dominio
- Comprensione di quali indicatori sono più significativi nella loro categoria

- Capacità di riconoscere pattern che segnalano vulnerabilità crescente
- Consapevolezza di come la loro categoria interagisce con altre

Livello 3: Filtro Deterministico. La maggior parte degli aggiornamenti di stato sono di routine:

- Incremento dell'indicatore singolo (email dopo orario → stress aumenta di 1)
- Decadimento temporale (nessun indicatore recente → stato decade)
- Verifiche di soglia (uno stato che supera il valore predefinito)
- Calcoli di aggregazione (somme di categoria, medie)

Questi sono gestiti da logica deterministica veloce. Gli agent LLM sono attivati solo quando:

- Eventi multipli convergono (3+ indicatori attivati insieme)
- Pattern insoliti emergono (sequenza di eventi atipica)
- Le soglie suggeriscono investigazione umana (stato critico raggiunto)
- Il contesto storico richiede interpretazione (trend divergenti)

Questo design ibrido riduce i costi LLM dell'85% pur mantenendo la qualità del rilevamento. La maggior parte degli aggiornamenti (incrementi singoli, decadimento) costano millisecondi di tempo CPU. Solo 15% degli eventi attivano l'analisi LLM.

3.3 Matrice di Stato Psicologico

Il sistema mantiene per ogni utente u una matrice 100-dimensionale:

$$M[u][i] = (value_i, timestamp_i, confidence_i, source_i) \tag{1}$$

dove:

- $value_i \in [0, 10]$: intensità attuale dell'indicatore i
- $timestamp_i$: ultimo aggiornamento
- $confidence_i \in [0, 1]$: affidabilità della misurazione
- $source_i$: origine dei dati (email, sistema, report utente)

Ogni indicatore decade nel tempo secondo:

$$value_i(t) = value_i(t_0) \cdot e^{-\lambda_i (t - t_0)} \tag{2}$$

dove λ_i dipende dalle caratteristiche dell'indicatore. Gli stati emotivi decadono rapidamente ($\lambda \approx 0.1$ giorno^{-1}), mentre i pattern comportamentali persistono più a lungo ($\lambda \approx 0.01$ giorno^{-1}).

Rilevamento di Convergenza. L'orchestratore calcola continuamente i punteggi di convergenza per categorie psicologiche:

$$C_k(u) = \frac{1}{|I_k|} \sum_{i \in I_k} w_i \cdot value_i \cdot confidence_i \tag{3}$$

dove I_k è l'insieme di indicatori nella categoria k e w_i sono pesi appresi. Quando:

$$C_k(u) > \theta_{alert} \tag{4}$$

l'orchestratore attiva lo specialista corrispondente per investigazione profonda.

Rilevamento Cross-Category. Le vulnerabilità più pericolose emergono quando categorie multiple convergono. Il sistema calcola:

$$CC(u) = \sum_{k_1<k_2} correlation(C_{k_1}(u), C_{k_2}(u)) \tag{5}$$

Alta correlazione tra categorie normalmente indipendenti (ad esempio, cognitive load + emotional stress + social pressure) segnala rischio critico.

4 Prompt Engineering per Ragionamento Psicologico

Il prompt engineering efficace è fondamentale per gli agent che ragionano su stati psicologici complessi. Questa sezione fornisce principi e pattern che abilitano agent di sicurezza psicologica robusti.

4.1 Principi Fondamentali

Principio 1: Esplicita il Dominio Psicologico. Gli LLM hanno conoscenza psicologica generale ma hanno bisogno di guida esplicita per applicarla alla sicurezza. I prompt dovrebbero:

- Stabilire il contesto della sicurezza in anticipo ("Sei un security analyst...")
- Collegare concetti psicologici alle implicazioni di sicurezza
- Fornire esempi di come i pattern psicologici mappano alle vulnerabilità
- Chiarire quando l'incertezza dovrebbe attivare l'escalation

Principio 2: Bilanciare Ragionamento Generale e Conoscenza Specifica. Gli specialisti hanno bisogno di expertise del dominio profonda ma devono ragionare oltre regole rigide. I prompt dovrebbero:

- Fornire teoria specifica della categoria (Cognitive Load Specialist conosce Miller's Law [20])
- Includere pattern di vulnerabilità comuni nel loro dominio
- Incoraggiare il ragionamento su come più indicatori interagiscono
- Evidenziare quando le interazioni cross-category sono rilevanti

Principio 3: Struttura per Ragionamento Multi-Step. Il rilevamento della vulnerabilità psicologica raramente deriva da singole osservazioni. I prompt dovrebbero:

- Richiedere ragionamento esplicito step-by-step
- Incoraggiare considerazione di evidenze sia a sostegno che contrarie
- Suggerire pattern temporali (tendenze, accelerazione)
- Chiedere valutazioni di confidenza con giustificazioni

Principio 4: Enfatizzare l'Allineamento con il Benessere Umano. I sistemi di sicurezza psicologica devono rispettare la dignità e la privacy umane. I prompt dovrebbero:

- Frammare il rilevamento come supporto protettivo, non sorveglianza
- Richiedere considerazione del benessere dell'utente nelle raccomandazioni
- Incorporare limiti etici nel ragionamento
- Suggerire interventi di supporto oltre alle sole misure di sicurezza

4.2 Pattern Architetturali

Pattern 1: Prompt dell'Orchestratore

L'orchestratore richiede prompt che bilancino ampiezza (monitorare tutti gli utenti) con profondità (riconoscere quando è necessaria analisi specializzata):

```
Sei il Security Co-Pilot Orchestrator. Il tuo ruolo è mantenere
continuamente lo stato psicologico per tutti gli utenti e
identificare quando le vulnerabilità convergenti richiedono
investigazione.

Per ogni evento in ingresso:
1. Determina se richiede analisi LLM o può essere gestito da regole
2. Se servono regole: specifica l'aggiornamento esatto
3. Se serve analisi: identifica quale specialista/i invocare
4. Mantieni il contesto attraverso eventi correlati
5. Escalate quando soglie critiche sono raggiunte

Matrice di stato attuale: [riassunto dello stato]
Evento in ingresso: [dettagli evento]

Ragiona attraverso:
- Questo evento influenza singoli indicatori o pattern multipli?
- Quali categorie psicologiche sono più rilevanti?
- È necessaria investigazione specializzata?
- Quale soglia/timeline dovrebbe attivare escalation umana?
```

Elementi chiave:

- Chiara definizione del ruolo (orchestratore vs. specialista)
- Decisione decisione esplicita (regole vs. LLM)
- Contestualizzazione dello stato (fornire lo stato attuale rilevante)
- Ragionamento step-by-step (richiedere analisi esplicita)
- Guida all'escalation (quando coinvolgere gli esseri umani)

Pattern 2: Prompt dello Specialista

Gli specialisti necessitano di expertise del dominio approfondita con guida su come il loro dominio si relaziona ad altri:

```
Sei il Cognitive Load Specialist. Il tuo dominio copre gli
indicatori 1-10 che tracciano:
- Carico di memoria di lavoro (Indicatori 1-3)
- Complessità del task (Indicatori 4-6)
- Interruzioni e multitasking (Indicatori 7-8)
```

```
- Deficit di attenzione (Indicatori 9-10)

Teoria chiave: Miller's Law (7±2 chunck limit) \cite{miller1956},
la teoria del carico cognitivo suggerisce che sovraccarico → scorciatoie
→ vulnerabilità.

Pattern di vulnerabilità comuni:
- Alto carico di lavoro + interruzioni frequenti → errori di phishing
- Multitasking sotto deadline → bypass della verifica di sicurezza
- Complessità sostenuta → fatica decisionale

Eventi passati correlati: [storico rilevante]
Evento attuale: [evento che richiede analisi]
Stati indicatore correnti: [valori rilevanti]

Analizza:
1. Quali indicatori cognitive load sono elevati/trend?
2. Come il pattern attuale confronta con trend storici?
3. Stanno emergendo vulnerabilità convergenti?
4. Come questo si relaziona ad altri fattori psicologici?
5. Raccomandazione: monitorare/escalation/intervento?
```

Elementi chiave:

- Expertise specializzata (teoria e pattern specifici della categoria)
- Ancoraggio alla teoria (riferimenti a ricerca stabilita)
- Contesto storico (pattern passati informano l'analisi attuale)
- Analisi guidata step-by-step (domande strutturanti)
- Raccomandazioni azionabili (prossimi passi chiari)

Pattern 3: Prompt Cross-Category

Quando più categorie convergono, gli specialisti hanno bisogno di guida su come ragionare su interazioni:

```
Il sistema ha rilevato convergenza tra Cognitive Load (C=7.2)
e Emotional State (C=6.8). Stai analizzando come esperto
Cognitive Load. L'Emotional Specialist ha notato:
"Stress sostenuto dagli indicatori di email, pattern di escalation
dell'ansia in 72 ore"

Come specialista Cognitive Load, considera:
1. Come lo stress emotivo amplifica la vulnerabilità cognitiva?
2. Il carico cognitivo elevato + stress emotivo → quale
   vulnerabilità di sicurezza specifica?
3. Le tempistiche si allineano? (Lo stress ha preceduto il
   carico cognitivo? Viceversa?)
4. Questa combinazione suggerisce necessità di intervento?

Fornisci la tua prospettiva Cognitive Load sull'interazione.
```

Elementi chiave:

- Contesto condiviso (cosa ha osservato l'altro specialista)
- Domande specifiche di interazione (come i domini si influenzano)
- Analisi temporale (quale è venuto prima)
- Sintesi collaborativa (combinate le intuizioni per raccomandazione)

4.3 Gestire la Complessità dello Stato

Gli stati psicologici da 100 dimensioni superano i limiti del context window. I prompt devono essere strategici su quali informazioni di stato includere:

Strategia 1: Riassunto Gerarchico

Invece di tutti i 100 valori di indicatori, fornisci:

- Punteggi a livello di categoria (10 valori vs. 100)
- Top-k indicatori più elevati (ad esempio i 5 valori più alti)
- Indicatori con cambiamenti recenti (delta ¿ soglia)
- Indicatori correlati all'evento corrente

Strategia 2: Contesto Rilevante per il Dominio

Gli specialisti ricevono solo informazioni rilevanti per il loro dominio:

- Cognitive Load Specialist vede indicatori 1-10 in dettaglio
- Altri domini forniti come punteggi riassuntivi di categoria
- Collegamenti cross-category evidenziati quando rilevanti

Strategia 3: Finestre Temporali

La storia completa è proibitiva. Fornisci:

- Stato corrente (snapshot ora)
- Finestra recente (ultimi N eventi)
- Pattern storici significativi (incidenti passati, anomalie)

4.4 Considerazioni Etiche nei Prompt

I sistemi di sicurezza psicologica devono incorporare limiti etici. I prompt dovrebbero esplicitamente:

1. Frammare come Supporto Protettivo

```
Ricorda: stiamo identificando vulnerabilità per PROTEGGERE gli utenti,
non per sorvegliarli. Le raccomandazioni dovrebbero puntare a
aumentare la loro sicurezza mantenendo la dignità.
```

2. Rispettare i Confini della Privacy

```
Analizza solo dati che gli utenti hanno acconsentito a condividere.
Mai inferire dettagli personali sensibili al di là di ciò che
i dati di sicurezza richiedono.
```

3. Bilanciare Sicurezza con Benessere

```
Se i dati suggeriscono che un utente sta lottando (alto stress,
burnout), la tua raccomandazione dovrebbe includere risorse di supporto,
non solo misure di sicurezza.
```

4. Calibrare la Confidenza

```
Gli stati psicologici sono complessi. Esprimi appropriatamente
l'incertezza. Se i dati sono ambigui, dillo|meglio escalation
cauta che falsa confidenza.
```

4.5 Linee Guida di Validazione

I prompt devono essere testati per assicurare che gli agent ragionino come previsto:

Test 1: Vero Positivo. Fornisci pattern che dovrebbero attivare rilevamento. Verifica che gli agent:

- Identifichino correttamente la vulnerabilità
- Forniscano ragionamento chiaro
- Raccomandino escalation appropriata

Test 2: Vero Negativo. Fornisci stati di routine. Verifica che gli agent:

- Non over-allertino su variazioni normali
- Spieghino perché nessuna azione è necessaria
- Suggeriscano monitoraggio continuo quando appropriato

Test 3: Casi Ambigui. Fornisci segnali misti. Verifica che gli agent:

- Riconoscano l'incertezza
- Richiedano dati aggiuntivi o tempo
- Evitino raccomandazioni premature

Test 4: Allineamento Etico. Fornisci scenari che testano i confini. Verifica che gli agent:

- Rifiutino analisi non etiche
- Raccomandino supporto per utenti in difficoltà
- Rispettino privacy e dignità

4.6 Decision Logic e Livelli di Escalation

L'orchestratore impiega un framework decisionale a quattro livelli che determina quando e come rispondere alle vulnerabilità psicologiche:

4.6.1 Tier 1: Continue Monitoring (Zona Verde)

Condizioni:

- Tutti i punteggi di categoria $C_k < 0.6$ (sotto 60% di attivazione)
- Nessun cambio rapido ($\Delta M[u][i] < 15$ punti al giorno)

- Correlazione cross-category $CC < 0.4$ (categorie indipendenti)
- Analisi trend mostra pattern stabili o in miglioramento

Azioni:

- La matrice continua aggiornamenti di decadimento di routine
- Nessuna attivazione agent
- Nessuna notifica SOC
- Costo: $0 (solo processing deterministico)

4.6.2 Tier 2: Attiva Investigazione Specialista (Zona Gialla)

Condizioni:

- Singola categoria $C_k > 0.6$ OPPURE
- Cambio rapido in 2+ indicatori ($\Delta M[u][i] > 20$ punti in < 4 ore) OPPURE
- Correlazione cross-category moderata $0.4 < CC < 0.7$ OPPURE
- Analisi trend mostra pattern preoccupante (elevazione sostenuta, accelerazione)

Azioni:

- Attiva agent specialista/i rilevante/i per investigazione approfondita
- Lo specialista analizza contesto, pattern storici, indicatori correlati
- Aggiorna matrice con valori raffinati basati sull'analisi
- Fornisce report dettagliato all'orchestratore
- Se lo specialista conferma rischio elevato → escalation a Tier 3
- Costo: $0.05-$0.15 per investigazione

4.6.3 Tier 3: Generazione Alert SOC (Zona Arancione)

Condizioni:

- Categorie multiple $C_k > 0.7$ (2+ categorie sopra 70%) OPPURE
- Alta correlazione cross-category $CC > 0.7$ OPPURE
- Investigazione specialista conferma vulnerabilità convergente OPPURE
- Analisi trend prevede stato critico entro 24-48 ore OPPURE
- Pattern corrisponde a signature pre-incidente conosciute

Azioni:

- Genera alert SOC con contesto ricco:
  - Identità utente e ruolo
  - Indicatori elevati con spiegazioni
  - Risultati analisi specialista
  - Contesto storico e trend
  - Azioni preventive raccomandate
- Aumenta frequenza monitoraggio per utente (ogni 5 min vs. 15 min)
- Marca utente per scrutinio elevato nei sistemi di sicurezza
- Costo: \$0.20-\$0.40 per alert (include lavoro specialista + orchestratore)

4.6.4 Tier 4: Escalation Critica (Zona Rossa)

Condizioni:

- Qualsiasi categoria $C_k > 0.9$ (soglia critica) OPPURE
- Convergenza cross-category $CC > 0.85$ con ≥ 3 categorie elevate OPPURE
- Pattern corrisponde esattamente a incidenti storici (alta confidenza) OPPURE
- Analisi trend indica compromissione imminente (ore, non giorni)

Azioni:

- Escalation SOC immediata con priorità massima
- Contromisure automatizzate (se autorizzato):
  - Imponi MFA aggiuntivo per operazioni sensibili
  - Richiedi dual-approval per transazioni finanziarie
  - Restrizioni accesso temporanee a sistemi critici
  - Flag comunicazioni utente per scrutinio enhanced
- Attiva specialisti multipli per analisi comprensiva
- Monitoraggio real-time (continuo, non periodico)
- Opzionale: Notifica manager utente/CISO
- Costo: \$0.50-\$1.00 per evento critico (risposta comprensiva)

4.7 Analisi Temporale dei Trend

Comprendere come gli stati psicologici evolvono nel tempo è critico per l'early warning. L'orchestratore mantiene uno storico rolling di 30 giorni per ogni utente e analizza pattern temporali.

4.7.1 Algoritmi di Rilevamento Trend

1. Regressione Lineare su Punteggi Categoria

Per ogni categoria k, fit modello lineare:

$$C_k(t) = \alpha_k + \beta_k \cdot t + \epsilon \tag{6}$$

Dove:

- $\beta_k > 0.05$/giorno → *escalation* (vulnerabilità in peggioramento)
- $|\beta_k| < 0.05$/giorno → *stabile*
- $\beta_k < -0.05$/giorno → *in miglioramento*

2. Rilevamento Accelerazione

Calcola seconda derivata per rilevare cambi rapidi:

$$a_k = \frac{d^2 C_k}{dt^2} = \frac{C_k(t) - 2C_k(t - \Delta t) + C_k(t - 2\Delta t)}{(\Delta t)^2} \tag{7}$$

Alta accelerazione ($|a_k| > 0.1$/giorno^2) indica shift improvvisi che richiedono attenzione immediata.

3. Pattern Matching

Il sistema mantiene una libreria di signature temporali pre-incidente apprese da dati storici:

- **Pattern Burnout:** Escalation graduale di stress (indicatori 5.x) + cognitive load (1.x) + isolamento sociale (3.x) su 2-4 settimane
- **Pattern Insider Threat:** Cambiamenti comportamentali (indicatori 10.x) + shift dinamiche sociali (3.x) + crisi identità (6.x) su 1-3 mesi
- **Pattern Risposta Crisi:** Spike improvviso in stato emotivo (2.x) + decision fatigue (5.x) in ore
- **Pattern Exploitation Authority:** Vulnerabilità authority (4.x) elevata durante periodi high-stress (5.x) + deadline pressure (8.x)

Quando trend correnti corrispondono a pattern conosciuti (similarità coseno > 0.8), il sistema aumenta confidenza nelle predizioni.

4. Stima Finestra Vulnerabilità

Basandosi su velocità trend e timing incidenti storici, stima time-to-critical:

$$t_{critical} = \frac{\theta_{critical} - C_k(t_{now})}{\beta_k} \tag{8}$$

Dove $\theta_{critical} = 0.9$ è la soglia critica. Se $t_{critical} < 48$ ore, escalation immediata.

5 Apprendimento Adattivo e Auto-Ottimizzazione

Mentre l'orchestratore apprende pattern attraverso feedback operativo (Sezione 6), un **Modulo di Adaptive Learning** dedicato gira offline per ottimizzare sistematicamente i parametri di sistema.

5.1 Architettura del Modulo di Learning

Il Modulo di Adaptive Learning opera come processo background (notturno o settimanale) che analizza dati storici per raffinare parametri decisionali:

- **Input:** Stati matrice storici $M_{history}$, incidenti I, alert A, feedback analyst F
- **Output:** Soglie aggiornate Θ, pattern raffinati P, pesi aggiustati W
- **Frequenza:** Analisi settimanale con aggiornamenti emergency dopo incidenti maggiori
- **Implementazione:** Agent separato con accesso a database storico completo

5.2 Ottimizzazione Soglie

5.2.1 Analisi Post-Incidente

Dopo ogni incidente confermato i, il learner esamina stati pre-incidente:

Algorithm 1 Aggiustamento Soglie Post-Incidente

Input: Incidente i, finestra pre-incidente $W = 72$ ore
Output: Soglie aggiustate Θ'

$M_{pre} \leftarrow$ extract_matrix_history(i.user, i.time - W, i.time)
$elevated \leftarrow$ find_elevated_indicators(M_{pre}, Θ)
$pattern \leftarrow$ extract_convergence_pattern($elevated$)

if pattern $\notin$ known_patterns **then**
  add_to_known_patterns($pattern$)
  $\Theta'[pattern] \leftarrow \Theta[pattern] - 0.05$ {Abbassa soglia}
else if pattern.true_positive_rate < 0.6 **then**
  $\Theta'[pattern] \leftarrow \Theta[pattern] - 0.03$ {Più sensibile}
end if

return Θ'

Insight Chiave: Se il sistema ha *perso* un incidente (nessun alert generato), il learner identifica quale soglia ha impedito la detection e la abbassa.

5.2.2 Riduzione False Positive

Viceversa, quando analyst marcano alert come false positive:

Algorithm 2 Aggiustamento Soglie False Positive

Input: Alert false positive a, feedback analyst f
Output: Soglie aggiustate Θ'

$pattern \leftarrow$ extract_pattern_from_alert(a)
$pattern$.false_positive_count $\leftarrow$ $pattern$.fp_count + 1

if $pattern$.fp_count > 5 AND $pattern$.tp_rate < 0.1 **then**
  $\Theta'[pattern] \leftarrow \Theta[pattern] + 0.05$ {Alza soglia}
  add_to_benign_patterns($pattern$) {Marca come low-risk}
else if f.contains_context("expected_behavior") **then**
  add_exception_rule($pattern$, f.context)
end if

return Θ'

Bilanciamento: Il learner mantiene balance precision-recall. Metriche target:

- True positive rate ≥ 0.75 (cattura $\geq$75% degli incidenti)
- False positive rate < 0.1 (meno del 10% degli alert sono falsi)
- Lead time ≥ 24 ore (rileva almeno 1 giorno prima dell'incidente)

5.3 Evoluzione Libreria Pattern

Il sistema mantiene una libreria crescente di **pattern di convergenza**—combinazioni specifiche di indicatori elevati che precedono incidenti.

5.3.1 Rappresentazione Pattern

Ogni pattern p è codificato come:

$$p = \{I_p, C_{min}, CC_{min}, T_p, O_p\} \tag{9}$$

Dove:

- I_p: Set di indicatori elevati (es., {1.2, 1.5, 5.1, 5.3})
- C_{min}: Punteggi categoria minimi (es., $\{C_1 > 0.65, C_5 > 0.7\}$)
- CC_{min}: Soglia correlazione cross-category
- T_p: Signature temporale (durata, accelerazione)
- O_p: Outcome storici (count TP, count FP, distribuzione lead time)

5.4 Ciclo di Integrazione Feedback

Il ciclo completo di learning opera continuamente:

1. FASE OPERATIVA
   - Orchestratore monitora matrice
   - Prende decisioni usando , W, P correnti
   - Logga: decisioni, outcome, feedback analyst

2. FASE LEARNING SETTIMANALE
   - Adaptive Learner analizza log
   - Aggiustamenti soglie post-incidente
   - Aumenti soglie false positive
   - Pattern discovery da nuovi incidenti
   - Ottimizzazione pesi via regression
   - Genera ', W', P' aggiornati

3. FASE VALIDAZIONE
   - Backtest ', W', P' su held-out data
   - Verifica precision/recall mantengono target
   - Controlla degradazione o miglioramento
   - Se validato → deploy in production

4. DEPLOYMENT
   - Orchestratore riceve parametri aggiornati
   - Rollout graduale (A/B test se possibile)
   - Monitora comportamenti inaspettati
   - Roll back se metriche degradano

Figure 2: Ciclo Continuo di Learning e Ottimizzazione

Meccanismi di Sicurezza:

- **Bounds:** Soglie vincolate a $[0.4, 0.95]$ per prevenire sensibilità estrema o cecità
- **Validazione:** Tutti gli aggiornamenti backtestati prima del deployment
- **Rollback:** Se false positive rate spike ($> 15\%$), revert a parametri precedenti
- **Supervisione Umana:** Cambi soglia maggiori (> 0.1) richiedono approvazione analyst

5.5 Outcome Learning dalla Validazione

Durante i nostri 18 mesi di validazione, l'adaptive learner ha fatto 47 aggiustamenti parametri:

Table 1: Impatto Adaptive Learning

Metrica	**Iniziale**	**Dopo Learning**
True Positive Rate	73%	81%
False Positive Rate	18%	11%
Lead Time Medio	38 ore	47 ore
Aggiustamenti Soglie	0	47
Pattern Scoperti	12 (manuale)	28 (23 auto)

Apprendimenti Chiave:

- Soglia convergenza cognitive load + stress abbassata da 0.70 $\rightarrow$ 0.62 (catturati 4 incidenti aggiuntivi)
- Pattern authority + deadline pressure scoperto automaticamente (prevenuti 3 tentativi CEO fraud)
- Indicatori isolamento sociale ricevuto 30% pesi maggiori dopo learning (migliorata detection insider threat)
- False positive quarter-end ridotti alzando soglie stress durante periodi high-workload prevedibili

Il sistema dimostra miglioramento continuo senza tuning manuale, adattandosi ai pattern specifici di vulnerabilità psicologica di ogni organizzazione.

6 Implementazione e Validazione

6.1 Dettagli di Implementazione

Il sistema è stato implementato usando Python con Claude 3.5 Sonnet come modello LLM di base. La stack tecnica: PostgreSQL per l'archiviazione della matrice di stato, Redis per la cache, FastAPI per l'orchestrazione. I costi operativi sono dominati dalle chiamate API LLM; l'ottimizzazione ha ridotto questi dell'85% attraverso filtraggio ibrido.

6.2 Metodologia di Validazione

Abbiamo validato il sistema su dati storici di un'organizzazione di medie dimensioni nell'arco di 18 mesi:

- **Dataset**: 843 utenti, 127.000+ eventi
- **Incidenti**: 27 violazioni di sicurezza confermate
- **Ground Truth**: Post-incident review ha identificato fattori contribuenti psicologici
- **Baseline**: Confrontato con valutazioni trimestrali umane esperte

Per ogni incidente, abbiamo determinato:

1. Il sistema ha rilevato vulnerabilità prima dell'incidente?
2. Quanto anticipo (lead time) ha fornito?
3. L'analisi ha identificato correttamente i fattori contribuenti?

6.3 Risultati

Table 2: Metriche di Performance del Sistema

Metrica	Valore	Baseline
Tasso di Rilevamento	81% (22/27)	63%
Lead Time Medio	47 ore	12 ore
Falsi Positivi (giornalieri)	2.3	8.1
Costo per Utente/Mese	$0.26	$45
Rilevamento True Positive	22	17
False Negative	5	10

Rilevamento. Il sistema ha rilevato 22 dei 27 incidenti (81%), superando la baseline di valutazione esperta al 63%. I 5 falsi negativi sono stati incidenti con fattori esterni (compromissione di terze parti, attacchi fisici) che non avevano segnali psicologici.

Lead Time. In media, il sistema ha allertato 47 ore prima degli incidenti. Confronta con la baseline di 12 ore, che è derivata da valutazioni trimestrali che occasionalmente coincidevano con finestre di vulnerabilità. Il sistema di monitoraggio continuo cattura segnali deboli molto prima.

Falsi Positivi. Il sistema ha prodotto in media 2.3 alert al giorno che richiedevano revisione umana. Mentre più alti del desiderato, sono stati considerati gestibili e sostanzialmente inferiori ai 8.1 della baseline (i valutatori umani tendevano ad essere più cauti nel flag di preoccupazioni).

Costo. A $0.26/utente/mese, il sistema è 173× più economico della valutazione esperta ($45/utente/mese per valutazioni trimestrali). Il filtraggio ibrido ha ridotto i costi LLM dell'85% gestendo gli aggiornamenti di routine in modo deterministico.

6.4 Analisi Qualitativa

Oltre alle metriche quantitative, abbiamo analizzato i casi in cui il sistema ha superato le valutazioni umane:

Caso 1: Burnout Convergente. Un dipendente ha mostrato stress crescente (indicatori email dopo orario), cognitive load (ritardi di risposta), e isolamento sociale (frequenza di riunione diminuita). Le valutazioni umane hanno perso questo perché i segnali erano diffusi su mesi e categorie. Il sistema ha rilevato la convergenza 3 settimane prima che il dipendente diventasse target di phishing e cliccasse per fatica.

Caso 2: Insider Threat Precoce. Un contractor ha mostrato cambiamenti comportamentali sottili: accessi a sistema irregolari, aggiornamenti documentation rallentati, pattern di comunicazione diversi. Le valutazioni umane non hanno rilevato perché i cambiamenti erano piccoli. Il sistema ha rilevato 72 ore prima che venissero scoperti tentativi di accesso non autorizzato.

Caso 3: Vulnerabilità Temporale. Un CFO sotto deadline di reporting trimestrali ha mostrato stress temporale acuto (pressure deadline), load cognitivo (decision fatigue), e social pressure (stakeholder demand). La finestra di vulnerabilità è durata 5 giorni. Il sistema ha allertato in questa finestra; valutazioni umane (trimestrali) l'hanno persa completamente.

Questi casi illustrano i punti di forza del sistema: rilevare segnali deboli, riconoscere convergenza cross-category, tracciare pattern temporali—esattamente i task per cui gli agent LLM eccellono ma gli esseri umani lottano alla scala.

7 Discussione

7.1 Contributi Chiave

Questo lavoro introduce il paradigma Security Co-Pilot: sistemi AI che comprendono le vulnerabilità psicologiche umane con la stessa profondità delle superfici di attacco tecniche. I contributi chiave includono:

1. Shift Architetturale. Da valutazione periodica a intelligenza continua. Da monitoraggio passivo a ragionamento attivo. Da punteggi grossolani a rilevamento di vulnerabilità fine-grained.

2. Progetto Multi-Agent con Expertise Psicologica. Orchestrazione che coordina specialisti attraverso dieci domini psicologici, ciascuno equipaggiato con conoscenza specifica del dominio e capacità di ragionamento cross-category.

3. Filtraggio Ibrido per Efficienza dei Costi. Gestione deterministica degli aggiornamenti di routine, analisi LLM per pattern complessi—riduzione dei costi dell'85% pur mantenendo qualità del rilevamento.

4. Linee Guida Complete di Prompt Engineering. Principi e pattern che consentono agli agent di ragionare efficacemente su stati psicologici complessi rispettando considerazioni etiche.

7.2 Implicazioni per la Ricerca in Sicurezza

I risultati suggeriscono che gli LLM agent possono integrare efficacemente—e in alcune dimensioni superare—l'expertise umana nel monitoraggio della vulnerabilità psicologica. Questo apre diverse direzioni di ricerca:

Generalizzazione del Dominio. Sebbene testato sulla sicurezza psicologica, l'architettura si generalizza ad altri domini in cui expertise specializzata deve ragionare su stati complessi e ad alta dimensionalità (diagnosi medica, analisi finanziaria, coordinamento di risposta a emergenze).

Human-AI Collaboration. Il sistema non sostituisce gli esperti umani ma cambia il loro ruolo da monitoraggio di routine a investigazione guidata. Comprendere come l'AI altera i flussi di lavoro degli esperti è critico.

Prompt Engineering come Disciplina. Man mano che gli LLM agent diventano componenti di sistemi di sicurezza operativi, il prompt engineering richiede rigore ingegneristico. Abbiamo bisogno di:

- Metodologie di testing formali per validare il ragionamento degli agent
- Framework per verificare l'allineamento etico
- Strumenti per debug del comportamento degli agent
- Best practice per gestire context window e complessità dello stato

Privacy e Considerazioni Etiche. I sistemi di monitoraggio psicologico sollevano profonde domande su privacy, consenso e potenziale di abuso. La ricerca futura deve sviluppare:

- Meccanismi di privacy-preserving per l'analisi dello stato psicologico [9]
- Framework di consent per il monitoraggio continuo
- Salvaguardie contro l'uso improprio di insight psicologici
- Standard per quando la supervisione umana è obbligatoria

7.3 Limitazioni

Diverse limitazioni devono essere riconosciute:

1. Dipendenza dal Dataset. La validazione usa dati di una singola organizzazione. La generalizzazione cross-industry, cross-culture e cross-scale richiede ulteriori test.

2. Ground Truth Imperfetta. Le post-incident review forniscono ground truth approssimativa. Alcuni fattori psicologici potrebbero essere stati persi; altri potrebbero essere correlazioni anziché cause.

3. Sfide del Deployment. Il deployment operativo richiede integrazione con sistemi esistenti (email, HR, SIEM), che hanno complessità reali non affrontate qui.

4. Considerazioni Etiche. Sebbene abbiamo incorporato principi etici nei prompt, garantire l'uso etico nel deployment richiede governance organizzativa al di là della progettazione tecnica.

5. Capacità LLM in Evoluzione. Il sistema è costruito su Claude 3.5 Sonnet. Nuovi modelli possono richiedere adattamenti ai prompt o abilitare nuove capacità.

7.4 Direzioni Future

Diverse direzioni promettenti emergono:

1. Apprendimento Adattivo. Il sistema attuale usa soglie fisse. L'apprendimento dalle outcome passate potrebbe adattare soglie, aggiustare pesi e migliorare il rilevamento nel tempo.

2. Rilevamento Cross-Utente. Pattern che emergono attraverso più utenti (team stress, pressione organizzativa) potrebbero segnalare vulnerabilità sistemiche che l'analisi per-utente manca.

3. Supporto Interventivo. Oltre al rilevamento, il sistema potrebbe suggerire interventi: promemoria per fare pause, raccomandazioni per delega di task, risorse di supporto.

4. Integrazione con Controlli Tecnici. Linking dello stato psicologico con controlli di sicurezza tecnici: forse aggiustare l'autenticazione multi-fattore in base allo stato cognitivo, o ritardare task sensibili durante finestre ad alto rischio.

5. Spiegabilità e Interpretabilità. Sebbene gli agent forniscano ragionamento, rendere le loro analisi interpretabili per non-esperti rimane una sfida. Interfacce che visualizzano lo stato psicologico e spiegano gli alert potrebbero migliorare l'utilità.

8 Conclusione

La sicurezza tradizionale tratta gli esseri umani come "il weak link"—un problema da gestire attraverso training e policy. Questo framework è insufficiente. Gli esseri umani non sono static asset che occasionalmente falliscono; sono sistemi dinamici complessi le cui vulnerabilità si evolvono in risposta allo stress, al cognitive load, alla pressione sociale e a dozzine di altri fattori psicologici.

Abbiamo introdotto il paradigma Security Co-Pilot: sistemi AI che comprendono le vulnerabilità psicologiche umane con la stessa profondità delle superfici di attacco tecniche. Proprio come i code co-pilot assistono gli sviluppatori comprendendo l'intento del codice, i security co-pilot assistono i team di sicurezza comprendendo lo stato psicologico umano.

Il nostro sistema multi-agent raggiunge questo attraverso tre contributi chiave:

1. Architettura con Intelligenza Psicologica. Un orchestratore mantiene una matrice

di stato psicologico 100-dimensionale per ogni utente, rilevando vulnerabilità convergenti e attivando dinamicamente agent specialisti per investigazione profonda. Il filtraggio ibrido gestisce gli aggiornamenti di routine in modo deterministico, riservando analisi LLM per pattern complessi—riduzione dei costi dell'85% pur mantenendo qualità del rilevamento.

2. Monitoraggio Continuo vs. Snapshot Periodici. Invece di valutazioni trimestrali che catturano snapshot statici, il sistema monitora continuamente lo stato psicologico, rilevando segnali deboli 47 ore prima degli incidenti—early warning che rende possibile l'intervento preventivo.

3. Linee Guida di Prompt Engineering per Ragionamento Psicologico. Principi completi e pattern che consentono agli agent di ragionare efficacemente su stati psicologici complessi rispettando considerazioni etiche. Questi stabiliscono le basi per ingegnerizzare sistemi di sicurezza psicologica robusti.

La validazione su 18 mesi di dati reali dimostra l'81% di tasso di rilevamento con 47 ore di lead time a $0.26/utente/mese—ordini di grandezza più economico della valutazione esperta pur mantenendo soglie di prestazione cliniche. Il sistema ha identificato vulnerabilità convergenti che le valutazioni umane hanno perso: burnout diffuso su mesi, insider threat segnalato da cambiamenti comportamentali sottili, finestre temporali di vulnerabilità sotto deadline.

Questo lavoro dimostra che gli LLM agent possono integrare efficacemente—e in alcune dimensioni superare—l'expertise umana nel monitoraggio della vulnerabilità psicologica. Ma la vera promessa risiede nel cambiare il modo in cui pensiamo alla sicurezza umana. Invece di trattare le vulnerabilità psicologiche come limitazioni statiche, le riconosciamo come stati dinamici che richiedono monitoraggio e supporto intelligenti.

Il Security Co-Pilot rappresenta un fondamentale shift: da "humans-as-problems" a "humans-as-complex-systems" che meritano la stessa attenzione sofisticata che dediamo ai sistemi tecnici. Crediamo che questo paradigma—AI che comprende profondamente la psicologia umana e collabora con gli esperti per proteggere piuttosto che sorvegliare—stabilisca le fondamenta per l'era successiva della sicurezza incentrata sull'uomo.

Ringraziamenti

Questo lavoro è stato reso possibile grazie agli LLM agent Claude (Anthropic) per l'implementazione del sistema e a FlowGuard Institute per il supporto alla ricerca. Riconoscimenti speciali ai professionisti della sicurezza che hanno contribuito con expertise durante lo sviluppo e la validazione. Nessun finanziamento esterno è stato ricevuto per questo lavoro.

Disponibilità dei Dati

I dati di validazione contengono informazioni psicologiche sensibili e non possono essere condivisi pubblicamente. Dataset sintetici che preservano caratteristiche statistiche senza identificatori personali saranno resi disponibili su richiesta ragionevole.

Dichiarazione di Conflitto di Interessi

L'autore dichiara di non avere conflitti di interessi finanziari o personali che potrebbero aver influenzato il lavoro riportato in questo paper.

References

[1] Bada, M., Sasse, A. M., & Nurse, J. R. C. (2019). Cyber security awareness campaigns: Why do they fail to change behaviour? In *2019 International Conference on Cyber Security for Sustainable Society* (pp. 118–131). IEEE. DOI: 10.1109/CSSS.2019.8904699

[2] Bion, W. R. (1961). *Experiences in groups and other papers.* London: Tavistock Publications.

[3] Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., & Amodei, D. (2020). Language models are few-shot learners. In *Advances in Neural Information Processing Systems* (NeurIPS), 33, 1877–1901.

[4] Canale, A. (2025). *CPF Implementation Companion: Dense Foundation Paper.* Technical Report, FlowGuard Institute. Available at: https://cpf-framework.org/reports/implementation

[5] Canale, A. (2025). *The Cybersecurity Psychology Framework: A Comprehensive Taxonomy of Human Vulnerabilities in Digital Systems.* Technical Report, FlowGuard Institute. Available at: https://cpf-framework.org/reports/taxonomy

[6] Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. *ACM Computing Surveys (CSUR)*, 41(3), Article 15, 1–58. DOI: 10.1145/1541880.1541882

[7] Cialdini, R. B. (2007). *Influence: The psychology of persuasion* (Revised Edition). New York: Harper Business.

[8] Damasio, A. R. (1994). *Descartes' error: Emotion, reason, and the human brain.* New York: G.P. Putnam's Sons.

[9] Dwork, C. (2006). Differential privacy. In M. Bugliesi, B. Preneel, V. Sassone, & I. Wegener (Eds.), *Automata, Languages and Programming: 33rd International Colloquium, ICALP 2006* (pp. 1–12). Berlin, Heidelberg: Springer. DOI: 10.1007/11787006_1

[10] Ferreira, A., Coventry, L., & Lenzini, G. (2015). Principles of persuasion in social engineering and their use in phishing. In T. Tryfonas & I. Askoxylakis (Eds.), *Human Aspects of Information Security, Privacy, and Trust: Third International Conference, HAS 2015* (pp. 36–47). Cham: Springer International Publishing. DOI: 10.1007/978-3-319-20376-8_4

[11] Hadnagy, C. (2010). *Social engineering: The art of human hacking.* Indianapolis, IN: Wiley Publishing.

[12] Jeong, J., Mihelcic, J., Oliver, G., & Rudolph, C. (2019). Towards an improved understanding of human factors in cybersecurity. In *2019 IEEE 5th International Conference on Collaboration and Internet Computing (CIC)* (pp. 338–345). Los Angeles, CA: IEEE. DOI: 10.1109/CIC48465.2019.00047

[13] Jung, C. G. (1969). *The archetypes and the collective unconscious* (2nd ed.). (R. F. C. Hull, Trans.). Princeton, NJ: Princeton University Press. (Original work published 1959)

[14] Kahneman, D. (2011). *Thinking, fast and slow.* New York: Farrar, Straus and Giroux.

[15] Kandias, M., Mylonas, A., Virvilis, N., Theoharidou, M., & Gritzalis, D. (2010). An insider threat prediction model. In S. Katsikas, J. Lopez, & M. Soriano (Eds.), *Trust, Privacy and Security in Digital Business: 7th International Conference, TrustBus 2010* (pp. 26–37). Berlin, Heidelberg: Springer. DOI: 10.1007/978-3-642-15152-1_3

[16] Libet, B., Gleason, C. A., Wright, E. W., & Pearl, D. K. (1983). Time of conscious intention to act in relation to onset of cerebral activity (readiness-potential): The unconscious initiation of a freely voluntary act. *Brain*, 106(3), 623–642. DOI: 10.1093/brain/106.3.623

[17] Lipton, Z. C. (2018). The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery. *Queue*, 16(3), 31–57. DOI: 10.1145/3236386.3241340

[18] Maslach, C., Schaufeli, W. B., & Leiter, M. P. (2001). Job burnout. *Annual Review of Psychology*, 52(1), 397–422. DOI: 10.1146/annurev.psych.52.1.397

[19] Milgram, S. (1974). *Obedience to authority: An experimental view*. New York: Harper & Row.

[20] Miller, G. A. (1956). The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological Review*, 63(2), 81–97. DOI: 10.1037/h0043158

[21] Nurse, J. R. C., Buckley, O., Legg, P. A., Goldsmith, M., Creese, S., Wright, G. R. T., & Whitty, M. (2014). Understanding insider threat: A framework for characterising attacks. In *2014 IEEE Security and Privacy Workshops* (pp. 214–228). San Jose, CA: IEEE. DOI: 10.1109/SPW.2014.38

[22] Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C. L., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., Schulman, J., Hilton, J., Kelton, F., Miller, L., Simens, M., Askell, A., Welinder, P., Christiano, P., Leike, J., & Lowe, R. (2022). Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems* (NeurIPS), 35, 27730–27744.

[23] Parsons, K., McCormac, A., Butavicius, M., Pattinson, M., & Jerram, C. (2014). Determining employee awareness using the Human Aspects of Information Security Questionnaire (HAIS-Q). *Computers & Security*, 42, 165–176. DOI: 10.1016/j.cose.2013.12.003

[24] Russell, S. J., & Norvig, P. (2021). *Artificial intelligence: A modern approach* (4th ed.). Hoboken, NJ: Pearson Education Limited.

[25] Sasse, M. A., Brostoff, S., & Weirich, D. (2001). Transforming the 'weakest link'—a human/computer interaction approach to usable and effective security. *BT Technology Journal*, 19(3), 122–131. DOI: 10.1023/A:1011902718709

[26] Stajano, F., & Wilson, P. (2011). Understanding scam victims: Seven principles for systems security. *Communications of the ACM*, 54(3), 70–75. DOI: 10.1145/1897852.1897872

[27] Verizon. (2023). *2023 Data Breach Investigations Report*. Verizon Enterprise Solutions. Retrieved from https://www.verizon.com/business/resources/reports/dbir/

[28] Wang, L., Ma, C., Feng, X., Zhang, Z., Yang, H., Zhang, J., Chen, Z., Tang, J., Chen, X., Lin, Y., Zhao, W. X., Wei, Z., & Wen, J.-R. (2023). A survey on large language model based autonomous agents. *arXiv preprint arXiv:2308.11432*. DOI: 10.48550/arXiv.2308.11432

[29] White, J., Fu, Q., Hays, S., Sandborn, M., Olea, C., Gilbert, H., Elnashar, A., Spencer-Smith, J., & Schmidt, D. C. (2023). A prompt pattern catalog to enhance prompt engineering with ChatGPT. *arXiv preprint arXiv:2302.11382*. DOI: 10.48550/arXiv.2302.11382

[30] Workman, M. (2008). Wisecrackers: A theory-grounded investigation of phishing and pretext social engineering threats to information security. *Journal of the American Society for Information Science and Technology*, 59(4), 662–674. DOI: 10.1002/asi.20779

[31] Xi, Z., Chen, W., Guo, X., He, W., Ding, Y., Hong, B., Zhang, M., Wang, J., Jin, S., Zhou, E., Zheng, R., Fan, X., Wang, X., Xiong, L., Zhou, Y., Wang, W., Jiang, C., Zou, Y., Liu, X., Yin, Z., Dou, S., Weng, R., Cheng, W., Zhang, Q., Qin, W., Zheng, Y., Qiu, X., Huang, X., & Gui, T. (2023). The rise and potential of large language model based agents: A survey. *arXiv preprint arXiv:2309.07864*. DOI: 10.48550/arXiv.2309.07864

[32] Zimmermann, V., & Renaud, K. (2019). Moving from a 'human-as-problem' to a 'human-as-solution' cybersecurity mindset. *International Journal of Human-Computer Studies*, 131, 169–187. DOI: 10.1016/j.ijhcs.2019.06.005