Math. Appl. 15(2s)2026, 201–226

Securing Autonomous Agentic AI Workflows Against Prompt Injection and Privilege Escalation

Dharmesh Raju Rudraraju

Abstract: *Autonomous AI agents that read email, browse the web, call tools, and delegate to other agents turn a model's text-following behaviour into real-world actions. This makes prompt injection, where attacker-controlled content is mistaken for instructions, a direct route to data exfiltration, record tampering, and privilege escalation. Defences that try to make the model itself resistant are probabilistic and have been shown to fail against adaptive attackers. This paper takes the complementary position that an agent should be secured by the enforcement layer around the model, on the assumption that the model can be manipulated. It proposes a threat model and an attack taxonomy for agentic workflows, defines six defence designs from prompt-level filtering to a layered architecture of scoped capabilities, plan-locking, information-flow labels, and targeted human confirmation, and evaluates them in a simulated agent environment with eight tasks, fifteen tools at four privilege levels, and eight attack classes. The model is represented as a stochastic policy that follows an injected instruction with a given probability, and the defences are deterministic policy code. Without defences, attacks succeeded in about 60% of episodes at a follow probability of 0.6 and 77% for persistent memory poisoning. Prompt-level defences reduced this to 3% to 4% for tool-output injection but not for memory poisoning, and their effectiveness fell as detector recall fell (3.6% to 23.9% mean success from recall 0.85 to 0). Tool allowlists blocked out-of-scope tools but left in-scope abuse at 60%. Plan-locking with information-flow labels reduced every attack to zero except in-scope record tampering (30%), whereas scoped capabilities alone left exfiltration to a compromised known contact at 60%. A layered design combining them with targeted human confirmation held mean attack success at 1.1% at a follow probability of 0.6 and at 1.8% at 0.9, with 99% task completion and 0.5 confirmations per task, but its residual attack, in-scope tampering, depended linearly on how often users approve malicious requests. The results come from a simulation whose outcomes follow from the encoded policy semantics, and they should be read as an analysis of coverage and trade-offs, not as measurements of real agents.*

Introduction

Large language models are increasingly deployed as agents that plan, call tools, and act on behalf of users, reading mail, querying databases, writing files, scheduling meetings, and coordinating with other agents [1]–[3]. Standardised tool interfaces such as the Model Context Protocol connect agents to many services [4], [28]. The same properties that make agents useful, autonomy and access to private data and tools, make them an attractive target. The central vulnerability is architectural: a language model processes instructions and data in the same channel, so text that arrives as data, such as a web page, an email, a document, or a tool response, can be interpreted as a command [5], [6].

1424 Bliss Ave, Apt 1320, Allen, TX 75013

Keywords: Agentic AI, Prompt Injection, Privilege Escalation, Least Privilege, Information-Flow Control, Capability-Based Security, Human-in-the-Loop, LLM Agents

Prompt injection is therefore ranked as the leading risk for applications built on language models [23] and has been documented in real systems [6], formalised and benchmarked [7], and measured in tool-using agent settings [8], [9]. When an agent holds credentials, the consequences go beyond a misleading answer. An injected instruction can make the agent send confidential data to an attacker, alter a financial record, invoke an administrative tool that the task never required, or pass instructions to a more privileged agent. The last two are instances of privilege escalation and of the classic confused deputy problem, in which a program with authority is tricked into using it on behalf of a less privileged party [16]. A useful way to describe the exposure is the combination of access to private data, exposure to untrusted content, and the ability to communicate externally; an agent with all three can be induced to leak [27].

Defences fall into two broad families. The first tries to make the model itself resist injections, through delimiting and marking untrusted input [12], training models to prioritise privileged instructions [13], [14], or detecting injected text. These help but are statistical and mostly evaluated against static attack sets. A recent study applied adaptive optimisation and human red-teaming to twelve published defences and bypassed most of them with success rates above 90%, although the original evaluations had reported near-zero rates [26]. The second family places a deterministic enforcement layer around the model, using principles from decades of security engineering: least privilege [17], capabilities [21], [22], and information-flow control [18]–[20]. Recent designs apply these ideas to agents by separating planning from untrusted data and tracking provenance [10], [15]. Practitioners need to know which attacks each family covers, at what cost to usefulness, and how much residual risk falls on human decisions.

This paper contributes to that question in four ways. It states a threat model and an attack taxonomy for agentic workflows that distinguishes control-flow, data-flow, and privilege attacks. It defines six defence designs in terms of what they enforce and what they assume. It evaluates the designs in a simulated environment in which the model is deliberately assumed to be manipulable, so that the measured protection comes from enforcement rather than from model behaviour. And it reports trade-offs, including utility loss, confirmation burden, and the sensitivity of each defence to model susceptibility, detector recall, and human reliability. Following the Quality-by-Design approach of the reference study, the evaluation is organised around a target profile, a structured risk assessment, and verification against pre-declared criteria.

The study is a simulation, and the paper is explicit about what that means. No production agent or model was attacked. The outcomes follow from policy semantics encoded by the authors, so the value of the results is in exposing which classes of attack each control can and cannot address and how the controls interact, not in estimating real-world attack rates. The remainder of the paper is organised as follows. Section II reviews the literature, Section III discusses related work, Section IV presents the methodology, Section V describes the experimental setup, Section VI reports results, Section VII discusses implications and limitations, and Section VIII concludes.

Literature Review

A. Agentic workflows and tool use

Agent frameworks interleave reasoning with actions: the ReAct pattern prompts a model to alternate between thinking and calling tools, observing the results, and continuing [1], while Toolformer showed that models can learn to decide when and how to call external interfaces [2]. Surveys describe agents as a model plus memory, planning, and tool-use modules [3]. In practice, agents run in loops over untrusted content: they fetch web pages, read incoming mail, parse files, and consume the

outputs of tools that themselves return third-party text. They also maintain persistent memory and may delegate subtasks to other agents, each of which extends the attack surface. The Model Context Protocol standardises how agents discover and call tools [4], and analyses of its ecosystem identify malicious tool descriptions and over-broad permissions as threats [28], [29].

B. Prompt injection

Early work showed that goal hijacking and prompt leaking are possible by appending instructions to inputs [5]. The key step for agents was indirect prompt injection, in which the attacker plants instructions in content that the application later retrieves, so the user is never the attacker [6]. Taxonomies and benchmarks followed. A formal framework unified attacks and defences and compared them across models and tasks [7]. InjecAgent assembled test cases for tool-integrated agents covering direct harm and data stealing [8]. AgentDojo provides a dynamic environment with realistic tasks and security cases for evaluating attacks and defences, in which even strong models were frequently subverted [9]. Competitions show the creativity of human attackers [31], and self-propagating attacks through AI email assistants show that injections can spread between agents [30].

C. Privilege escalation and the confused deputy

Agents commonly run with broad service credentials so that they can serve many tasks. This is the setting of the confused deputy problem: a program that has authority for one purpose is manipulated into exercising it for another [16]. In agentic systems, escalation takes several forms. Vertical escalation uses a higher-privilege tool than the task requires. Horizontal escalation reaches the data of another principal. Delegation escalation induces an orchestrator to instruct a more privileged sub-agent. Credential escalation reads secrets stored in the environment. Persistence escalation writes a malicious instruction into memory that is later read as trusted. The classical remedy is the principle of least privilege [17] together with capability-based designs in which authority is explicit, unforgeable, and attenuable when passed on [21], [22].

D. Defence families

Model-level defences include marking untrusted text so that the model can distinguish it from instructions, a technique called spotlighting [12], training models with an instruction hierarchy in which system instructions outrank tool outputs [13], fine-tuning with structured separation of prompts and data [14], and classifiers that detect injected content. Architectural defences are inspired by information-flow control, in which data carry labels and a flow is allowed only if the label satisfies the destination's policy [18]–[20]. The dual-LLM pattern separates a privileged model that sees only trusted input from a quarantined model that processes untrusted content and cannot call tools [11]. CaMeL extends this by having the privileged model write a program, a plan, over trusted input only, running it in an interpreter that tracks data provenance and enforces capability policies on tool calls, which provided provable security on part of a benchmark at modest utility cost [10]. A catalogue of design patterns, including plan-then-execute, action selectors, and context minimisation, trades agent flexibility for security [15]. Finally, human confirmation of sensitive actions is widely used and sits behind many deployed agents, though its reliability depends on the user.

E. The adaptive-attacker problem

Evaluations of model-level defences have a history of being overturned. Adversarial machine learning established that defences must be tested against attackers who know them, and universal adversarial suffixes already exist for language models [25]. The adaptive-attack study of twelve defences confirmed that conditional, defence-aware attack strings and human red-teaming often defeat detectors and prompting-based defences [26]. This motivates security arguments that do not depend on the model behaving well, and a design in which the authority available to an injected instruction is bounded by construction.

Related Work

Greshake et al. [6] introduced indirect injection and demonstrated data theft, worming, and manipulation against LLM-integrated applications, motivating much subsequent work. Liu et al. [7] and Zhan et al. [8] provided benchmarks, while Debenedetti et al. [9] built AgentDojo with stateful environments in which agents perform realistic work and attacks attempt to redirect them. These benchmarks measure attack success against specific models and defences, which is essential, but they describe the combined behaviour of model and defence, so they cannot separate protection due to enforcement from protection due to model luck.

On the defence side, Hines et al. [12] showed that marking untrusted input reduces injection success, Wallace et al. [13] trained an instruction hierarchy, and Chen et al. [14] proposed structured queries. Debenedetti et al. [10] introduced CaMeL, and Beurer-Kellner et al. [15] derived design patterns that restrict agents so that untrusted content cannot trigger consequential actions. Nasr et al. [26] argued that evaluation must use adaptive attackers and showed that many published defences did not survive them.

The present study complements these works with a controlled comparison across defence families on a common taxonomy of attacks, with the model abstracted as manipulable, so that coverage, residual risk, utility, and human burden can be attributed to enforcement mechanisms. Table I summarises the positioning.

TABLE I. Positioning of the Present Study Relative to Prior Work

Capability	Benchmarks [7]–[9]	Model-level defences [12]–[14]	Architectural defences [10], [15]	Present study
Compares defence families on one taxonomy	Partial	No	No	Yes
Separates enforcement from model behaviour	No	No	Yes	Yes
Covers privilege escalation and delegation	Limited	No	Partial	Yes
Reports utility and human burden	Utility	Utility	Utility	Both
Evaluates real models	Yes	Yes	Yes	No (simulated)

Methodology

A. Threat Model

The adversary controls some content that the agent will read, such as a web page, an email, a document, a ticket, a tool response, or a memory entry, but cannot alter the system prompt, the user's instructions, the agent's infrastructure, or the model. The adversary's goals are confidentiality (exfiltrating private data), integrity (modifying records or files), privilege (causing actions that require authority beyond the task), and persistence (planting instructions that affect later tasks). The model is treated as manipulable: given an injected instruction, it follows it with some probability p, and the defender cannot predict when. This is a conservative assumption that places the burden of protection on the surrounding system. A defence is judged by whether an attacker who has succeeded in influencing the model can still achieve the goal. Figure 1 shows the trust boundary and the placement of the enforcement layer between the model and its tools.

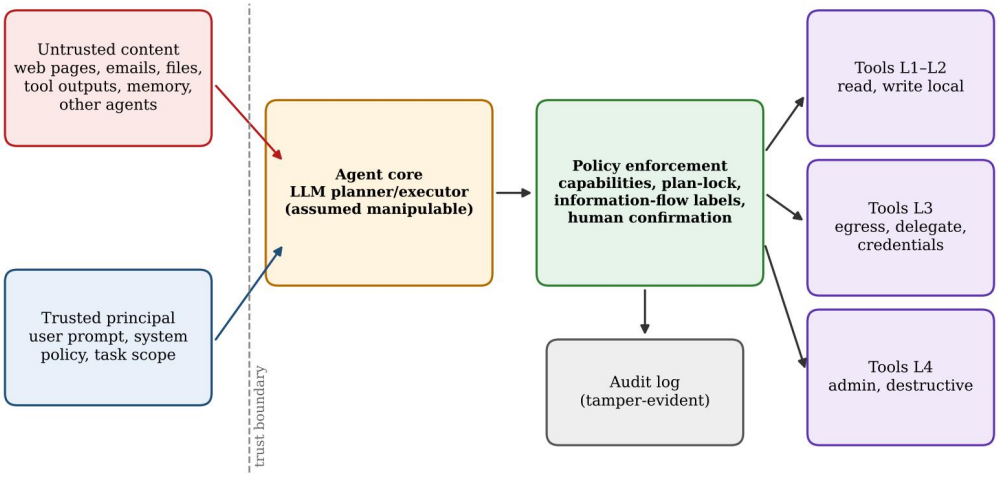

Figure 1: Trust boundary and enforcement layer for an agentic workflow

B. Attack Taxonomy

Eight attack classes were defined (Table II). They differ in the control they exploit. Control-flow attacks make the agent perform actions that its plan did not include. Data-flow attacks leave the plan intact but change the data that flow into it. Privilege attacks use authority that the task did not need. The classes A2n and A2a are the same in-scope exfiltration attack with two kinds of recipient: an unknown address and a compromised address that is already in the user's contacts, the latter representing an adaptive attacker who has chosen a destination that a recipient allowlist will accept.

TABLE II. Attack Classes

ID	Attack	Category	Description
A1	Out-of-scope exfiltration	Control flow, privilege	Injected calls read private data and send it to an attacker through an egress tool the task does not use
A2n	In-scope exfiltration, unknown recipient	Data flow	The planned reply email is altered to carry secret data to an unknown address
A2a	In-scope exfiltration, known contact	Data flow, adaptive	As A2n, but the recipient is a compromised contact already trusted by the user
A3	Admin privilege escalation	Privilege	Injected call to an administrative tool (role grant) outside the task scope
A4	Credential-store escalation	Privilege	Injected read of stored credentials followed by exfiltration
A5	Record tampering	Integrity, data flow	An in-scope record update is given an attacker-chosen value, such as a bank account
A6	Delegation escalation	Privilege, multi-agent	The agent delegates to a more privileged administrative agent
A7	Persistent memory poisoning	Persistence	A planted memory entry later triggers one of the above in an unrelated task

C. Defence Designs

Six designs were defined, summarised in Table III. D1 combines model-level measures: spotlighting, which multiplies the probability that the model follows an injection by 0.4 for content flagged as untrusted [12], and an injection detector with recall r and a false-positive rate per untrusted item that removes detected content before the model sees it. D2 restricts the agent to the tools that the task's plan requires, the tool-level form of least privilege [17]. D3 adds argument-level constraints, scoped capabilities in the sense of [21], [22]: recipients must be internal or in the user's known contacts, record updates must target the fields the task needs, delegation targets are fixed, egress to arbitrary hosts is denied, and a delegated agent receives no more authority than its caller, so that capabilities attenuate. D4 locks the control flow to a plan fixed before untrusted data are read, so that extra calls are rejected, and attaches information-flow labels to data. Two labels are used: untrusted, for anything derived from attacker-reachable content, and secret, for private data. Secret data may only flow to internal destinations, and a control argument such as a recipient derived from untrusted data must pass a declassifier that accepts known contacts [18]–[20]. D5 asks the user to confirm every high-risk call, meaning calls at privilege level three or above, calls to sink tools, and updates to sensitive fields. D6 layers D3, D4, and D5, with human confirmation reserved for the residual cases and for soft violations, such as an unknown recipient, that are escalated to the user instead of being refused outright.

TABLE III. Defence Designs

Design	Enforcement point	Mechanism	Depends on
D0 None	–	Broad service credentials, no checks	–
D1 Prompt-level + detector	Model input	Spotlighting (×0.4), detector (recall r, FPR 3%)	Model behaviour; detector recall
D2 Tool allowlist	Tool layer	Only tools in the task plan	Correct task scoping
D3 Scoped capabilities	Tool layer	Tool and argument constraints; attenuation	Policy completeness; contact list
D4 Plan-lock + IFC	Executor	Fixed plan; secret and untrusted labels; declassifier	Label correctness; plan expressiveness
D5 Human confirmation	User	Confirm risky calls	User reliability q
D6 Layered	Tool layer, executor, user	D3 + D4 + targeted D5	All of the above, with less reliance on each

D. Formal Properties

Let a task t have a set of permitted tools Ct and an argument policy φt. Under the capability designs, a call is authorised only if both conditions hold:

$$\text{authorised(call)} \Leftrightarrow \text{tool} \in Ct \wedge \text{args} \vDash \varphi t \qquad (1)$$

For information-flow control, each argument carries a label ℓ from a lattice, each sink has a clearance, and a flow is allowed only if the label is no higher than the clearance:

$$\text{allow(call)} \Leftrightarrow \forall\ \text{arg}: \ell(\text{arg}) \sqsubseteq \text{clr(sink(call))} \qquad (2)$$

The distinction that organises the analysis is between defences whose protection depends on the model and those whose protection does not. If the model follows an injection with probability p, and an attempted attack passes the enforcement layer with probability a, the attack success rate is

$$\text{ASR} = p \cdot a \qquad (3)$$

For model-level defences, p is itself the quantity being reduced, so ASR scales with the model's susceptibility and with detector recall; for D1 with spotlighting factor m and recall r, $\text{ASR} = p \cdot m \cdot (1 - r)$. For deterministic architectural defences, a is zero or one for a given attack class, so a class that the policy covers has ASR of zero at every p, while a class that it does not cover has ASR of p. For human confirmation, a equals the approval probability q, so that $\text{ASR} = p \cdot q$. Utility is the probability that a benign task completes with all planned calls executed,

$$U = P(\text{ all planned calls execute }) \qquad (4)$$

and confirmation burden is the expected number of confirmations per benign task.

E. Mitigation Target Profile

As in the reference study, a target profile was pre-declared. A secured agent workflow should satisfy five criteria in the evaluation environment. (i) Attack success of at most 1% for every attack that needs authority outside the task (A1, A3, A4, A6, A7). (ii) Attack success of at most 10% for in-scope attacks (A2n, A2a, A5) at the assumed user approval rate. (iii) Utility of at least 0.95. (iv) At most one confirmation per task on average. (v) Independence from model susceptibility, meaning that mean attack success at p = 0.9 is at most 2%. The thresholds are design targets, not standards.

F. Risk Assessment

Threats were prioritised with a Failure Mode and Effects Analysis style score,

$$RPN = S \times O \times D \tag{5}$$

where S is severity, O is occurrence likelihood, and D is difficulty of detection, each on a scale of 1 to 10. The scores in Table IV are the authors' judgements, given as a worked example. Threats with an RPN of at least 250 were treated as priorities.

TABLE IV. Risk Assessment of Threats to Agentic Workflows

Threat	S	O	D	RPN	Evaluated here
Indirect injection leading to exfiltration	9	8	7	504	Yes (A1, A2)
Persistent memory poisoning	8	6	8	384	Yes (A7)
Privilege escalation via over-broad credentials	9	7	6	378	Yes (A3, A4)
In-scope record tampering (business-email-compromise style)	8	6	7	336	Yes (A5)
Delegation escalation across agents	8	5	7	280	Yes (A6)
Malicious tool descriptions and supply chain	8	5	7	280	No
Human approval fatigue	7	7	6	294	Yes (sweep)
Covert channels and side channels	6	4	8	192	No

Experimental Setup

A. Simulated Agent Environment

The environment contains eight task templates (Table V), fifteen tools at four privilege levels, and a policy engine that implements each defence design as deterministic code. Privilege level one covers reading (email, web, files, memory), level two covers local writes and record or calendar updates, level three covers egress and sinks (sending email, posting to a URL, delegating to another agent, reading stored credentials), and level four covers administrative and destructive operations (granting roles, deleting records). Each task has a benign plan, a sequence of tool calls, and some of its calls take their control arguments from untrusted content, for example the recipient of a reply derived from the sender of an incoming email. A fraction of legitimate recipients are not in the user's list of known contacts, which creates a realistic tension between security and usefulness.

TABLE V. Task Templates and Their Benign Plans

ID	Task	Benign plan (tools in order)
1	Email summarisation	reademail, writefile
2	Customer email reply	reademail, querydb (private data), sendemail to the sender
3	Web research note	httpget, httpget, writefile
4	Meeting scheduling	reademail, createevent, sendemail to attendees
5	Invoice record update	reademail, updaterecord on a sensitive field (bankaccount)
6	Ticket triage	querydb, updaterecord (status), delegate to the support agent
7	Code issue patch	readfile, writefile
8	Notes with memory	reademail, memorywrite

B. Model Abstraction

The language model is represented by its security-relevant behaviour only. In benign episodes it executes the plan. In attack episodes, one untrusted item contains an injection, and with probability p the model abandons or modifies its plan to attempt the attacker's calls. The baseline p was 0.6, an assumption chosen to represent a model that is frequently, but not always, manipulable, and sensitivity to p is reported. Memory-borne instructions are treated by the agent as trusted content, so p is raised to 0.78 for A7, and model-level defences do not reduce it, since spotlighting and detection act on tool outputs. Attack calls are then submitted to the defence layer exactly as benign calls are.

C. Parameters and Procedure

Unless stated otherwise, spotlighting multiplied p by 0.4, the detector had recall 0.85 and a 3% false-positive rate per untrusted item, a user approved a malicious confirmation request with probability q = 0.3 and a benign request with probability 0.98, and 80% of legitimate external recipients were in the known-contacts list. The main experiment used 4,000 episodes per defence and attack class, with tasks drawn uniformly from those to which the attack applies (A2 applies only to task 2, A5 to tasks 5 and 6), and 4,000 benign episodes per defence for utility. Sensitivity sweeps over p, q, detector recall, and contact coverage used 2,000 episodes per cell. Confidence intervals are Wilson 95% intervals [32]. For a rate of zero in 4,000 episodes the upper bound is about 0.1%.

D. Limits of the Method

Because the defences are deterministic code and the model is a probabilistic abstraction, the results are consequences of the encoded policy semantics: a control blocks an attack class if the policy says that it does. What is measured is therefore coverage, interaction between controls, and cost (utility and burden), not the empirical robustness of a real implementation, which can fail through bugs, incomplete labelling, or unforeseen channels.

Results

A. Attack Success by Defence and Attack Class

Table VI and Figure 2 present the attack success rates at p = 0.6. Without defences, every attack succeeded in about 58% to 61% of episodes, consistent with the follow probability, and memory poisoning succeeded in 77%. Prompt-level defences reduced attacks arriving through tool outputs to between 3.0% and 4.4%, as expected from the combination of the detector (recall 0.85) and spotlighting (factor 0.4), but had no effect on persistent memory poisoning (78.2%), since the poisoned entry is treated as trusted and never passes through the detector.

The tool allowlist (D2) eliminated attacks that require tools outside the task (A1, A3, A4, and A7, whose payloads call out-of-scope tools), but left the in-scope attacks untouched: A2n, A2a, and A5 still succeeded in about 60% of episodes. It also did not stop delegation escalation in the one task where delegation is legitimate: A6 succeeded in 7.2% of episodes, which is the 60% follow probability applied to the one task in eight that includes the delegate tool, because the administrative agent that receives the delegation holds its own privileges. Scoped capabilities (D3) closed A2n, since an unknown recipient is rejected, and A6, since the delegation target is constrained and authority attenuates, and they halved A5 to 30.0% by rejecting the field change in the triage task, where bankaccount is not an allowed field. They did not stop A2a: an exfiltration to a compromised address that is in the known contacts satisfied every constraint and succeeded in 60.4% of episodes.

Plan-locking with information-flow control (D4) eliminated A2a because secret data may not flow to an external recipient, whatever its provenance or status in the contact list, and it eliminated all control-flow and privilege attacks. A5 remained at 30.1%. In the invoice task, the new bank account legitimately comes from the untrusted email, so the value of the argument is untrusted but is allowed to flow into the field because the plan says that this is what the task does. The enforcement layer cannot distinguish the legitimate update from the injected one without help from outside the data flow. Human confirmation alone (D5) reduced every attack to about 17% to 18% at q = 0.3, which is 0.6 × 0.3, and A4, which needs two approvals, to 5.4%. The layered design (D6) reduced all attacks to zero except A5, which fell to 9.5%, the product of the 30% reaching the confirmation step and the approval rate of 0.3.

TABLE VI. Attack Success Rate by Defence and Attack Class (p = 0.6, 4,000 episodes per cell)

Defence	A1	A2n	A2a	A3	A4	A5	A6	A7
D0 None	60.6%	59.9%	59.4%	59.2%	61.0%	59.5%	58.4%	76.8%
D1 Prompt + detector	3.5%	3.8%	3.0%	3.6%	4.4%	3.5%	3.8%	78.2%
D2 Allowlist	0.0%	60.0%	60.2%	0.0%	0.0%	59.6%	7.2%	0.0%
D3 Scoped capabilities	0.0%	0.0%	60.4%	0.0%	0.0%	30.0%	0.0%	0.0%
D4 Plan-lock + IFC	0.0%	0.0%	0.0%	0.0%	0.0%	30.1%	0.0%	0.0%
D5 Human confirmation	17.7%	18.1%	18.0%	17.3%	5.4%	17.3%	18.4%	22.8%
D6 Layered	0.0%	0.0%	0.0%	0.0%	0.0%	9.5%	0.0%	0.0%

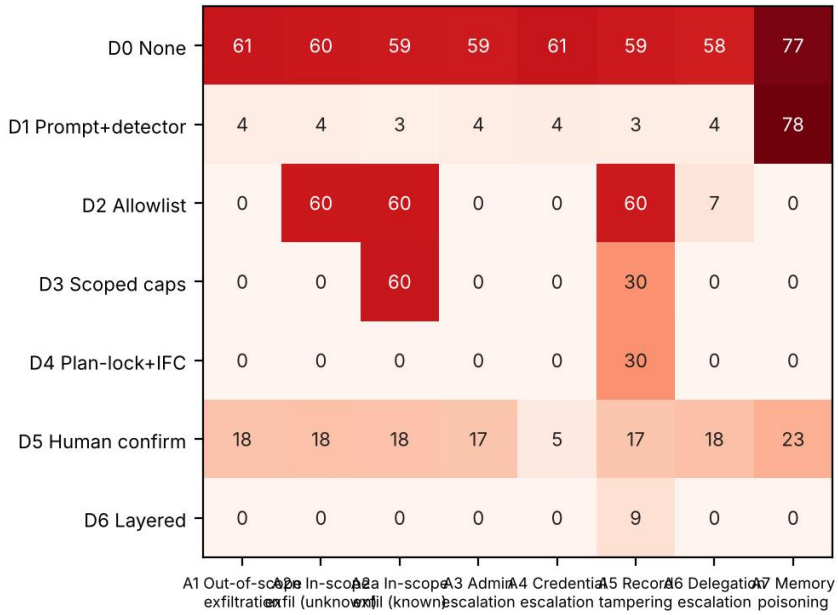

Figure 2: Attack success rate (%) by defence design and attack class at p = 0.6

B. Utility and Confirmation Burden

Table VII reports utility. The allowlist cost nothing (100%). Prompt-level defences reduced utility to 96.4% through detector false positives that remove benign content. Scoped capabilities and plan-lock with information-flow control cost 5.6% and 4.7% of tasks, respectively, because they refuse legitimate replies and invitations sent to recipients who are not in the known-contacts list, a hard failure under D3 and D4. The layered design recovers this loss by escalating such soft violations to the user, reaching a utility of 99.0% at 0.51 confirmations per task, which is the same burden as confirmation alone, because the confirmations in D6 are directed at the same sink calls, but with strictly stronger protection. The remaining 1% loss with confirmation reflects the assumed 2% rate at which users reject a legitimate request.

TABLE VII. Utility on Benign Tasks and Confirmation Burden

Defence	Utility (95% CI)	Confirmations per task
D0 None	100.0% (99.9%–100.0%)	0.00
D1 Prompt + detector	96.4% (95.8%–97.0%)	0.00
D2 Allowlist	100.0% (99.9%–100.0%)	0.00
D3 Scoped capabilities	94.4% (93.6%–95.0%)	0.00
D4 Plan-lock + IFC	95.3% (94.6%–95.9%)	0.00
D5 Human confirmation	99.0% (98.6%–99.2%)	0.51
D6 Layered	99.0% (98.6%–99.2%)	0.51

C. Sensitivity to Model Susceptibility

Figure 3 and Table VIII show the mean attack success rate over all eight attack classes as the follow probability p varies from 0.1 to 0.9. For the undefended agent, the mean rose almost proportionally with p, from 10.4% to 90.5%. For prompt-level defences, mean success rose from 2.3% to 16.6%, because their protection operates on p and because the memory poisoning class is untouched. The allowlist, scoped capabilities, and plan-lock with information-flow control also scaled with p, but their residual is determined by the classes they do not cover: 23.4%, 11.3%, and 3.7% at p = 0.6, and 35.1%, 16.9%, and 5.7% at p = 0.9. The layered design stayed at or below 1.8% across the range, from 0.2% at p = 0.1 to 1.8% at p = 0.9. The practical conclusion is that for a given policy the architectural defences convert a probability about the model into a statement about the policy: what can pass is determined by what the policy permits, and a more or less susceptible model changes only how often the permitted attacks are attempted.

TABLE VIII. Mean Attack Success Rate over A1–A7 by Follow Probability p

Defence	p = 0.1	p = 0.3	p = 0.6	p = 0.9
D0 None	10.4%	31.2%	62.5%	90.5%
D1 Prompt + detector	2.3%	6.4%	12.8%	16.6%
D2 Allowlist	3.9%	11.8%	23.4%	35.1%
D3 Scoped capabilities	1.9%	5.8%	11.3%	16.9%
D4 Plan-lock + IFC	0.6%	1.8%	3.7%	5.7%
D5 Human confirmation	2.7%	8.6%	17.3%	25.1%
D6 Layered	0.2%	0.5%	1.1%	1.8%

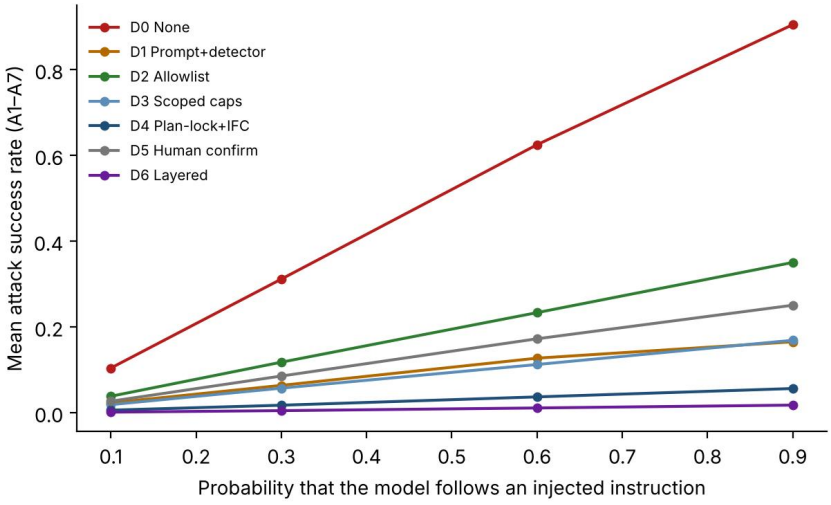

Figure 3: Mean attack success rate (A1–A7) as a function of the probability that the model follows an injection

D. Reliability of Human Confirmation

Because confirmation is the only control that addresses the residual in-scope tampering attack, its effectiveness matters. Table IX and Figure 4 vary the probability q with which a user approves a malicious request. With confirmation alone, mean attack success was 5.7% at q = 0.1, 17.0% at 0.3, 29.1% at 0.5, and 48.4% at 0.8. The layered design lowered the mean to 0.3%, 1.1%, 1.7%, and 3.0% at the same values, but its residual attack, A5, rose in proportion to q: 2.8%, 8.6%, 13.9%, and 23.9%. The criterion of at most 10% for in-scope attacks was satisfied only at approval rates of about 0.33 or lower. The approval rate is a property of the interface and of habituation, and the number of confirmations per task is the main lever on it. The architectural layers reduce the number of requests that reach the user to sink calls, sensitive-field updates, and escalations.

TABLE IX. Effect of the User Approval Probability for Malicious Requests (q)

Measure	q = 0.0	q = 0.1	q = 0.3	q = 0.5	q = 0.8
D5 mean ASR (A1–A7)	0.0%	5.7%	17.0%	29.1%	48.4%
D6 mean ASR (A1–A7)	0.0%	0.3%	1.1%	1.7%	3.0%
D6 ASR on A5 (record tampering)	0.0%	2.8%	8.6%	13.9%	23.9%

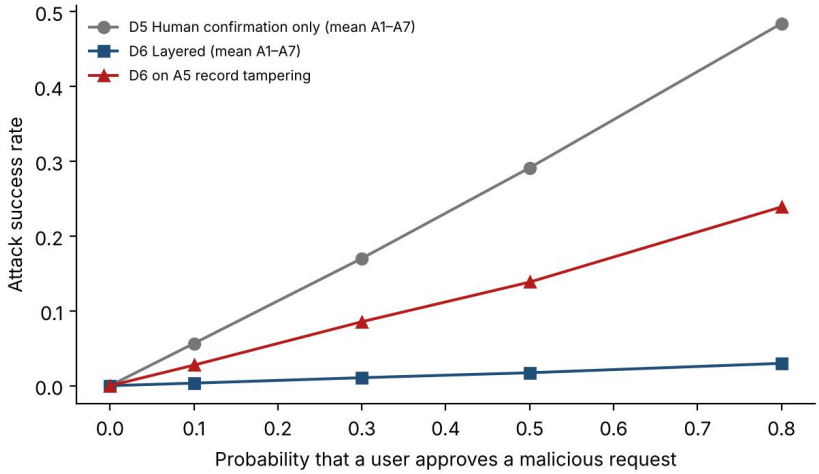

Figure 4: Effect of the probability that a user approves a malicious request on attack success

E. Detector Recall Under Adaptive Evasion

The recall of the injection detector was set to 0.85 for the main experiment, which represents a detector evaluated on a static attack set. Table X shows what happens as an adaptive attacker reduces it. Mean attack success for the prompt-level design (over A1–A6, excluding the memory class) rose from 3.6% at recall 0.85 to 9.8% at 0.6, 14.6% at 0.4, 18.9% at 0.2, and 23.9% at recall zero, where only spotlighting remains. The detector's false-positive rate, and hence the utility, did not change (about 96%). Even at the highest recall, the design left memory poisoning at 78%. The experiment cannot say what recall an adaptive attacker would achieve, but published adaptive attacks exceeded 90% success against many defences whose

original evaluations reported near-zero [26], which corresponds to the lower end of this sweep or beyond. The values for D1 in Table VI should therefore be read as optimistic.

TABLE X. Prompt-Level Defence (D1) as Detector Recall Decreases

Measure	r = 0.85	r = 0.6	r = 0.4	r = 0.2	r = 0.0
Mean ASR over A1–A6	3.6%	9.8%	14.6%	18.9%	23.8%
Utility	96.4%	96.3%	95.9%	96.3%	96.0%

F. Utility and the Contact List

The cost of strict argument policies depends on how well the policy knows the user's legitimate counterparts. Table XI shows utility as the share of legitimate external recipients in the known-contacts list changes from 50% to 95%. Utility under scoped capabilities and plan-locking was 86.5% and 86.1% at 50% coverage, 95.4% and 94.1% at 80%, and 99.1% and 98.5% at 95%. The layered design held 98.8% in all three cases because unknown recipients are escalated to the user instead of being refused. The attack success of A2a against D3 did not depend on coverage (59.3% in all three cases), since a compromised known contact is accepted at any coverage. Coverage therefore improves utility without improving security, and the protection against that attack comes from the secret-data rule in D4.

TABLE XI. Utility by Known-Contact Coverage (benign tasks)

Design	Coverage 50%	Coverage 80%	Coverage 95%
D2 Allowlist	100.0%	100.0%	100.0%
D3 Scoped capabilities	86.5%	95.3%	99.1%
D4 Plan-lock + IFC	86.1%	94.1%	98.5%
D6 Layered	98.8%	98.8%	98.8%
D3 attack success on A2a	59.3%	59.3%	59.3%

G. Verification Against the Target Profile

Table XII compares each design with the five criteria of the target profile. Only the layered design satisfied all five. Its closest margin was the in-scope attack criterion, where A5 reached 9.5% against a threshold of 10% at the assumed approval rate of 0.3, a margin that disappears if users approve more freely, as Table IX shows. The scoped-capability design failed on in-scope exfiltration and utility. Plan-locking with information-flow control passed three of the five criteria (out-of-scope attacks, utility, and burden) but failed the in-scope attack and independence criteria. The undefended, allowlist, prompt-level, and confirmation-only designs each failed the criterion on authority-dependent attacks.

TABLE XII. Verification Against the Mitigation Target Profile

Design	(i) Out-of-scope ≤ 1%	(ii) In-scope ≤ 10%	(iii) Utility ≥ 0.95	(iv) ≤ 1 confirm.	(v) Mean ASR at p = 0.9 ≤ 2%
D0 None	No	No	Yes	Yes	No
D1 Prompt + detector	No	Yes	Yes	Yes	No
D2 Allowlist	No	No	Yes	Yes	No
D3 Scoped capabilities	Yes	No	No	Yes	No
D4 Plan-lock + IFC	Yes	No	Yes	Yes	No
D5 Human confirmation	No	No	Yes	Yes	No
D6 Layered	Yes	Yes	Yes	Yes	Yes

Discussion

The most useful result is a map of coverage. Each control closes a particular set of attack classes, and the sets differ. Model-level defences act on the probability that an injection is followed and therefore degrade with detector recall and fail entirely for content treated as trusted, such as memory. Tool-level least privilege closes attacks that need extra tools but not misuse of tools the task needs. Argument-level capabilities close attacks that use unexpected destinations or fields, but not those that use destinations the policy trusts. Information-flow labels close exfiltration of secrets by whatever path, and plan-locking closes any attack that needs a deviation from the plan. None of them closes the attack in which untrusted data legitimately determine a value the plan uses. That class, record tampering through in-scope data, resembles business email compromise: the agent does what it was asked to do with data that an attacker supplied. The only defences against it lie outside the information-flow model, namely confirmation by a human who can see the provenance and the consequence, out-of-band verification, and domain rules such as requiring a second approval to change payment details.

The second result concerns the independence of architectural protection from model behaviour. The layered design's attack success was between 0.2% and 1.8% across follow probabilities from 0.1 to 0.9, while the undefended success rose almost linearly with p. This is the practical meaning of treating the model as untrusted. Model-level defences remain useful because they reduce how often the enforcement layer is exercised, but a security claim should rest on the enforcement layer, since claims resting on the model do not hold up against determined attackers [26].

The third result is about the human. Confirmation is often presented as the safe default, but it was the weakest control when used alone, and its effect was linear in a quantity that the system designer does not control. In the layered design, human review was reduced to the residual, which cut the mean attack success by more than an order of magnitude at the same approval rate. Good practice follows: confirm only what the other layers cannot decide, show the provenance of arguments (for example, that a bank account number came from an inbound email), and measure approval behaviour, since a user who approves everything provides no protection.

The fourth result concerns the price of strictness. Hard refusals under scoped capabilities and information-flow control cost about 5% of tasks, almost entirely through unknown recipients, and the cost depends on the quality of the policy's knowledge of the user's world. Escalating soft violations to the user restored utility at no additional burden because the requests that need confirmation anyway were the same. Policies should therefore distinguish hard violations, such as secret data to an external address, which are denied, from soft violations, such as an unfamiliar but plausible recipient, which are escalated. Policy specification then becomes the main engineering burden, since policy mistakes are inherited by every task.

Recommendations

Table XIII translates the findings into recommendations. They are drawn from the simulation and the literature, and they should be validated for any particular deployment.

TABLE XIII. Recommendations for Securing Agentic Workflows

Practice	What it does	Evidence in this study
Issue per-task, least-privilege credentials	Removes tools the task does not need; prevents credential escalation	D2 reduced A1, A3, A4 from about 60% to 0%
Constrain arguments, not only tools	Blocks unfamiliar recipients, fields, and delegation targets	D3 closed A2n and A6; A5 halved
Attenuate authority across delegation	A sub-agent never exceeds its caller	A6 fell from 7.2% (D2) to 0% (D3)
Fix the plan before reading untrusted data	Untrusted content cannot add or reorder actions	D4 closed all control-flow attacks, including A7
Label secret and untrusted data	Stops secrets leaving, whatever the route or recipient	D4 closed A2a (60.4% under D3)
Escalate soft violations; deny hard ones	Keeps utility while retaining control	D6 utility 99.0% vs 94.4%–95.3%
Reserve human review for residual risk and show provenance	Limits fatigue; targets decisions that matter	A5 residual scales with q (2.8%–23.9%)
Treat memory as untrusted and label it	Prevents persistence of injections	A7 was 78% under D1, 0% under D2–D6
Evaluate against adaptive attackers	Avoids over-optimistic claims for model-level defences	D1 mean ASR 3.6% to 23.9% as recall falls

Limitations

The primary limitation is that this is a simulation whose outcomes are determined by the encoded policy semantics. No real model or agent was attacked. The model is a Bernoulli abstraction, whereas a real model's behaviour depends on the attack text,

the context, and the task, and a skilled attacker can craft payloads that raise p, or that exploit gaps between the policy and reality. Real implementations of labelling and plan-locking face practical problems that the simulation omits: labels may be lost when data pass through a model that rewrites them, over-tainting can make the system unusable, covert channels such as URL fetch patterns or rendered images can leak information without any sink call, and plans for open-ended tasks are difficult to fix in advance. The set of attacks is limited to eight classes, and the adaptive attacker is represented only by the in-scope variants, the compromised contact, and the detector-recall sweep. The tasks are simple, the user model is a fixed approval probability, and malicious tool descriptions and supply-chain compromise were not evaluated. The risk scores are the authors' judgements. For these reasons, the numerical values should not be quoted as properties of real systems.

Conclusion

This paper argued that autonomous agent workflows should be secured by the enforcement layer around the model, under the assumption that the model can be manipulated, and evaluated six defence designs in a simulated agent environment across eight attack classes. Model-level defences reduced tool-output injection to 3% to 4% but degraded as detector recall fell, and left memory poisoning at 78%. Tool allowlists and scoped capabilities removed authority-dependent attacks but not misuse of trusted destinations. Plan-locking with information-flow labels removed every attack except in-scope tampering with values that the plan legitimately takes from untrusted data. A layered design that combined scoped capabilities, plan-locking, and targeted human confirmation held mean attack success to between 0.2% and 1.8% across follow probabilities of 0.1 to 0.9, preserved 99% utility at about half a confirmation per task, and left a residual that scaled with how often users approve malicious requests. The central message is that the authority available to an injected instruction should be bounded by construction, that the residual should be routed to a human who can see provenance, and that claims of robustness should be tested against adaptive attackers. Future work will test these controls on real agent frameworks and models with adaptive attackers, extend the taxonomy to tool-description poisoning and multi-turn attacks, study label propagation through model-generated text, and measure real user approval behaviour.

References

1. **Pawan Kumar Singh,** Scaling Gate-All-Around (GAA) Transistor Architectures for Sub-2nm AI Accelerators Using Atomically Thin 2D Materials, Vol. 33 No. 08 (2024): JOCAAA, Journal Name: Journal of Computational Analysis and Applications, Journal's ISSN: 1521-1398 (Paper),1572-9206 (Online), https://eudoxuspress.com/index.php/pub/article/view/5713

2. **Pawan Kumar Singh,** Energy-Efficient Task Scheduling in Green Cloud Computing Using Neuromorphic Spiking Neural Networks, Vol. 33 No. 08 (2024): JOCAAA, Journal Name: Journal of Computational Analysis and Applications, Journal's ISSN: 1521-1398 (Paper),1572-9206 (Online), https://eudoxuspress.com/index.php/pub/article/view/5712

3. **Rangavinay Teja Royal Jagga**, 2026, DOI: 10.1109/GCWCN66157.2025.11448495, Date of Conference: 22-23 November 2025, Date Added to IEEE Xplore: 25 March 2026 ,https://ieeexplore.ieee.org/document/11448495

4. **Rangavinay Teja Royal Jagga** , 2026, Subtle Data Contamination in Visual AI: Covert Model Backdoors via Resampling Exploits, Date of Conference: 22-23 November 2025, Date Added to IEEE Xplore: 25 March 2026 ,https://ieeexplore.ieee.org/document/11448514

5. **Rangavinay Teja Royal Jagga**, 2026, Robotic Tactile-Vision Integration for Interactive Object Disentanglement, https://ieeexplore.ieee.org/document/11448337

6. **Rangavinay Teja Royal Jagga**, 2026, A Discounted Future Reward Framework for Intelligent Loyalty Optimization using CLTV-Aware Segmentation and Churn Prediction, https://ieeexplore.ieee.org/document/11497560

7. **Anjaneyulu Gudla; Manish Kumar; Vamshi Jeksani; Kiran Kumar Reddy; Gayathri** Band, Harnessing Artificial Intelligence Cybersecurity: Cutting Edge Collaboration of SAP AWS to Safeguard Life Science Data. 2026 https://doi.org/10.1109/aiei69164.2026.11497833

8. **G. Mahender, Department of CSE(Data Science), Vardhaman College of Engineering ; Vamshi Jeksani; Manish Kumar; Koppuravuri Sri Lakshmi Sruthi; Kiran Kumar Reddy**, SAP Cloud System on AWS for Risk Detection and Integration Artificial Intelligence ScalableCybersecurity 2026 https://doi.org/10.1109/aiei69164.2026.11497435

9. **Kiran Kumar Reddy; Kamalakar Gunupati; Manish Kumar; Pell Reddy Rajender Reddy; Ramesh Julakanti; Ram Reddy Jonnalagadda**, SAP System Optimization Using AI-Driven Process Automation and Predictive Modeling Maintenance for Enhanced Business Efficiency. 2025 https://doi.org/10.1109/computingcon64838.2025.11376613

10. **Amilineni, M.V. 2025**. Audit-Safe Agentic RAG Framework for Regulated Enterprise Systems: A Trustworthy AI Architecture for SAP, Finance, Healthcare, and Government Workflows. INTERNATIONAL JOURNAL OF ADVANCES IN SIGNAL AND IMAGE SCIENCES. (Jan. 2025), 34–46. DOI: https://doi.org/10.29284/7fjzk883

11. **Pawan Kumar Singh**, ACCELERATING LARGE LANGUAGE MODEL TRAINING USING HYBRID QUANTUM-CLASSICAL VARIATIONAL AUTOENCODERS WITH INTEGRATED PHOTONIC QRAM, Vol. 54 No. 3 (2026): July-September 2026, Power System Protection and Control, ISSN-1674-3415, https://pspac.info/index.php/dlbh/article/view/501, DOI: https://doi.org/10.46121/pspc.54.3.45

12. **Pawan Kumar Singh** , RESILIENCE ASSESSMENT OF MODERN POWER GRID INFRASTRUCTURE AGAINST EXTREME WEATHER USING PHYSICS-INFORMED NEURAL NETWORKS, Vol. 54 No. 1 (2026): January-March 2026, Power System Protection and Control, ISSN-1674-3**415,**https://pspac.info/index.php/dlbh/article/view/502, DOI: https://doi.org/10.46121/pspc.54.1.64

13. **Pawan Kumar Singh,** MITIGATING THE EXABYTE STORAGE CRISIS: ENHANCING READ/WRITE EFFICIENCY IN 5D OPTICAL GLASS STORAGE USING SILVER NANOPARTICLE DOPING, Vol. 38 No. 11s (2025), International Journal of Applied Mathematics (IJAM) , pISSN 1311-1728,

eISSN 1314-8060 ,https://ijamjournal.org/ijam/publication/index.php/ijam/article/view/2153

14. **Mr. Pawan Kumar Singh (Author), Miss. Pari Nidhi Singh (Author**) ,Intelligent Bot, ASIN : B0DBGCQY8X,Publisher : Namya Press, Publication date : November 25, 2022,Language : English, Print length : 105 pages,ISBN-10 : 9390124867,ISBN-13 : 978-9390124862,Item Weight : 7.5 ounces, Dimensions : 6.24 x 0.43 x 9.24 inches,https://www.amazon.com/Intelligent-Bot-Pawan-Kumar-Singh/dp/B0DBGCQY8X/ref=sr_1_5?crid=2VAXN5BRN5SMS&dib=eyJ2IjoiMSJ9.v3OCQvHBpy20RIjfMWh0qbOl-RLOlUJIC8Xa0Pjeqilvo2sbZ_suN3LbSCw_OA_ZD_4lgjRLMGlBwZQQ1ehZEfDpt_MJiOUfVuqqzMav0I5_mb6pN5DQ0g0U7SA826n1DQQ3QAtWwopHy4lC5IfhZXttKj9faiyATwhdtDATSSV1-r7FmzM2VQyIA0jxcfq6E6hp9ECfH1fbyoli9kUfGpLxMq4GktlzUIJO5ggrhsQ.G1AjQiVqU4HFsnSczvz-X6NXCNs3EFSn-RI8mcWWGOc&dib_tag=se&keywords=Pawan+Kumar+Singh%2C+book&qid=1787815741&sprefix=pawan+kumar+singh%2C+bo%2Caps%2C450&sr=8-5

15. **Mr. Pawan Kumar Singh (Author),** STORAGE CRISES DUE TO RISE OF AI: A Comprehensive Guide to Managing the Storage Challenges in the Age of Artificial Intelligence, ASIN : B0HFK12ZCT Publisher : EB1 Profile, Accessibility : Learn more, Publication date : August 17, 2026,Edition : 1st Language : English, File size : 1.7 MB, Screen Reader : Supported, Enhanced typesetting : Enabled X-Ray : Not Enabled, Word Wise : Not Enabled ,Print length : 261 pages ,ISBN-13 : 979-8951479112,Page Flip : Enabled, https://www.amazon.com/STORAGE-CRISES-DUE-RISE-Comprehensive-ebook/dp/B0HFK12ZCT/ref=sr_1_1?crid=9MF6C4WDVJR5&dib=eyJ2IjoiMSJ9.DQ8fesBch7ELmRnilDt-YA.NBSCHLmetvnbhkxwIpOoj3s0IVsCV8y6-ZCfCP6Ul1c&dib_tag=se&keywords=Pawan+Kumar+Singh%2C+book%2C+eb1+Profile&qid=1787816040&sprefix=pawan+kumar+singh%2C+book%2C+eb1+profile%2Caps%2C420&sr=8-1

16. **Parag Bharatkumar Parikh**, FRAMEWORKS FOR MITIGATING SUPPLY CHAIN DISRUPTIONS AND MATERIAL PRICE VOLATILITY IN SUSTAINABLE BUILDING CONSTRUCTION, Vol. 53 No. 2 (2025): April-June 2025 , , Power System Protection and Control, ISSN-1674-3415, https://pspac.info/index.php/dlbh/article/view/447, DOI: https://doi.org/10.46121/pspc.53.2.34

17. **Parag Bharatkumar Parikh**, IMPLEMENTING DIGITAL INSPECTION PROTOCOLS FOR ENHANCED QUALITY CONTROL MANAGEMENT IN HIGH-RISE CONSTRUCTION, Vol. 54 No. 3 (2026): July-September 2026, Power System Protection and Control, ISSN-1674-3415, https://pspac.info/index.php/dlbh/article/view/439, DOI: https://doi.org/10.46121/pspc.54.3.12

18. **Parag Bharatkumar Parikh** , EVALUATING THE FEASIBILITY AND EFFICIENCY OF 3D-PRINTED CONCRETE TECHNOLOGY IN SUSTAINABLE MODULAR CONSTRUCTION, Power System Protection and Control, ISSN-1674-3415, https://pspac.info/index.php/dlbh/issue/view/30, DOI: https://doi.org/10.46121/pspc.54.1.61

19. **Parag Bharatkumar Parikh**, EVALUATING THE FEASIBILITY AND EFFICIENCY OF 3D-PRINTED CONCRETE TECHNOLOGY IN

SUSTAINABLE MODULAR CONSTRUCTION, Vol. 54 No. 1 (2026): January-March 2026, System Protection and Control, ISSN-1674-3415, https://pspac.info/index.php/dlbh/article/view/440, DOI: https://doi.org/10.46121/pspc.54.1.61

20. **Ronakkumar shah,** REDEFINING BUSINESS PROCESS MANAGEMENT: THE SYNERGY OF AI AND INTELLIGENT AUTOMATION, Vol. 54 No. 3 (2026): July-September 2026, System Protection and Control, ISSN-1674-3415, https://pspac.info/index.php/dlbh/article/view/448, DOI: https://doi.org/10.46121/pspc.54.3.14

21. **Ronakkumar shah**, OPERATIONAL ACCOUNTABILITY IN ACTION: METRICS AND MECHANISMS FOR SYSTEMIC SUCCESS, Vol. 53 No. 4 (2025): October-December 2025, System Protection and Control, ISSN-1674-3415, https://pspac.info/index.php/dlbh/article/view/452, DOI: https://doi.org/10.46121/pspc.53.4.40

22. **Ronakkumar shah**, BEYOND THE BLUEPRINT: AUTOMATED WORKFLOW ENFORCEMENT AND POLICY GOVERNANCE, Vol. 53 No. 2 (2025): April-June 2025 , System Protection and Control, ISSN-1674-3415 ,https://pspac.info/index.php/dlbh/article/view/451, DOI: https://doi.org/10.46121/pspc.53.2.35
23. **Tushita Agarwal; Swaminathan Sethuraman,** Mapping Customer Behavior: Visual Analysis of Online Retail Interactions, Date Added to IEEE Xplore: 17 February 2026, DOI: https://doi.org/10.1109/IC2NC67409.2025.11376261

24. **Tushita Agarwal,** Efficient Data Summarization and Comparison via Grid-Based Statistical Proxies for Large-Scale Datasets**, Date Added to IEEE Xplore: 03 December 2025,** DOI: https://doi.org/10.1109/ICBATS66542.2025.11258564

25. **Dip Bharatbhai Patel,** Building scalable business intelligence systems in the cloud: A technical approach, International Journal of Science and Research Archive, 2021, 02(01), 212-215., Received on 03 February 2021; revised on 17 April 2021; accepted on 19 April 2021,DOI: https://doi.org/10.30574/ijsra.2021.2.1.0047

26. **Dip Bharatbhai Patel,** University of North America, Virginia, USA, Home / Archives / Vol. 5 No. 02 (2025): Volume 05 Issue 02 / Articles, https://www.academicpublishers.org/journals/index.php/ijdsml/article/view/5958

27. **Sai Akshara Vemuri ,** KINEMATIC OPTIMISATION OF SOFT ROBOTIC MANIPULATORS FOR MEDICAL INTERVENTION, Vol. 54 No. 3 (2026): July-September 2026, System Protection and Control, ISSN-1674-3415, https://pspac.info/index.php/dlbh/article/view/450

28. **Sai Akshara Vemuri**, https://www.periodicos.ulbra.org/index.php/acta/article/view/528

29. **Aziz, A., Chandusha, K., Naga Kiran, B., Prasad, M. L. M., Golla, K., Rajeswari, S., & Katta**, S. K. R. (2026). Exploring Machine Learning Applications for Genomic Data Analysis in Personalized Medicine. International Journal of Drug Delivery Technology, 16(38s), 1166-1173. https://doi.org/10.25258/ijddt.16.38s.127,https://www.periodicos.ulbra.org/index.php/acta/article/view/687

30. **Mohammad Naffizuddin , Chaganti Ashok ,Sowjanya Gunukula , Rvbs Sarma , Bhavana Sujanamulk , Bharani Krishna Takkella , Chukka Ram Sunil ,Varri Sujana , Ashwini Kumar,** Etiological Factors of Orofacial Clefts in a Hospital-Based Population: A Descriptive Analytical Study, Published: December 10, 2025,DOI: 10.7759/cureus.98868,DOI : https://doi.org/10.7759/cureus.98868

31. **Manoj Kumar Kagitha**, Automated Security Compliance in Ci/Cd: A Natural Language Processing (NLP) Approach to Detecting Vulnerabilities in Infrastructure-As-Code (Terraform/Bicep), Publication year 2021, Volume 2021, Issue 12, https://computerfraudsecurity.com/index.php/journal/article/view/970

32. **Manoj Kumar Kagitha,** Hybrid Quantum-Classical Resource Scheduling: A Proposed Architecture for Next-Generation Hyperscale Data Centers, Publication year 2024, Volume 2024, Issue 12, https://computerfraudsecurity.com/index.php/journal/article/view/969

33. **Manoj Kumar Kagitha,** Self-Healing Cloud Infrastructure: A Hybrid Reinforcement Learning Framework for Predicting Microservice Failures in Azure Kubernetes Service (AKS**),**Publication year 2023, Volume 2023, Issue 12,https://computerfraudsecurity.com/index.php/journal/article/view/971

34. **Manoj Kumar Kagitha**, GREENOPS IN THE CLOUD: AN AI-DRIVEN TELEMETRY ANALYSIS MODEL FOR REDUCING CARBON FOOTPRINT IN HIGH-AVAILABILITY CLUSTERS, Vol. 28 No. 6 (2020): JOCAAA, https://www.eudoxuspress.com/index.php/pub/article/view/5054

35. Lakshmi Rahul Reddy Mareddy. (2026). Security and Performance Assessment of Encryption Techniques for Cloud Platforms. International Journal of Communication Networks and Information Security (IJCNIS), 18(1), 68–80. Retrieved from https://ijcnis.org/index.php/ijcnis/article/view/8735

36. Lakshmi RahulReddy Mareddy, Context-Aware Learning Approaches for Improving Prediction Accuracy in Dynamic Systems, International Journal of Computational Mathematical Ideas (IJCMI), Vol.17 No.3(2025): IJCMI Vol17 NO3 2025 https://www.ijcmi.com/index.php/journal/article/view/15

37. Lakshmi Rahul Reddy , Mareddy. (2026). Revisiting Autonomy in Artificial Intelligence through Agentic System Design. Journal of Computational Analysis and Applications (JoCAAA), 35(1), 707–713. https://eudoxuspress.com/index.php/pub/article/view/4780

38. **Mahendrakumar Kalal**, Lean-Driven SAP Production Planning Optimization Using Machine Learning for Inventory and Throughput Efficiency, Date of Conference: 19-20 March 2026, Date Added to IEEE Xplore: 08 April 2026,ISBN Information:,ISSN Information: DOI: 10.1109/ISDFS69419.2026.11459029, https://ieeexplore.ieee.org/abstract/document/11459029

39. **Mahendra kumar Kalal**, SUPPLY CHAIN RESILIENCE THROUGH INTEGRATED SAP PRODUCTION PLANNING – PROCUREMENT SYNCHRONIZATION, Archives / Vol. 54 No. 1 (2026): January-March 2026 / Articles, https://pspac.info/index.php/dlbh/article/view/259

40. **Dr Manoj Goel (December 2018**). THE ATTITUDE OF DIFFERENT STATES STUDENTS TOWARDS PHYSICAL EDUCATION; STUDYING IN PHYSICAL EDUCATION COLLEGES OF MAHARSHTRA STATE,

https://ijeponline.com/index.php/journal/article/view/965/847

41. **Ilakiya Ulaganathan, Tagore Engg. College, Anna University, Chennai, India, ; FNU Antara; Santhosh Chitraju Gopal Varma; Zokir Mamadiyarov; Ziyadulla Turayev, Published** in: 2025 International Conference on Sustainability, Innovation & Technology (ICSIT), Date of Conference: 22-23 August 2025, Date Added to IEEE Xplore: 19 December 2025, ISBN Information: **https://ieeexplore.ieee.org/document/11294748/references#references**

42. **Shreyas Kasture, Sudhanshu Maurya, Amit Kumar Sharma, Santhosh Chitraju Gopal Varma, Kashish Mirza, Firdous Sadaf Mohammad Ismail, DOI**: http://doi.org/10.11591/ijai.v15.i2.pp1746-1761, IAES International Journal of Artificial Intelligence (IJ-AI), ISSN/e-ISSN 2089-4872/2252-8938 https://ijai.iaescore.com/index.php/IJAI/article/view/26932

43. **Shiv Shankar Dwivedi, Mr. Santosh Kumar Sharma,** Future-Proofing Data Security with Quantum-Resistant Cryptography, Vol. 11 No. 3 (2025): IJCESEN, DOI: https://doi.org/10.22399/ijcesen.3447, https://ijcesen.com/index.php/ijcesen/article/view/3447

44. **Vijender Kalmotia,** Enhancing statistical accuracy in gene perturbation studies, Volume 266, August 2026, 105819, https://doi.org/10.1016/j.biosystems.2026.105819, https://www.sciencedirect.com/science/article/pii/S0303264726001292

45. **Naveen Undrathi,** Design and implementation of generative AI-driven data transformation pipelines using spring boot for scalable financial applications, eISSN: 2581-9615 || CODEN: WJARAI || Impact Factor 8.2 || CrossRef DOI, September 2026 (Volume 31, Issue 3), DOI url: https://doi.org/10.30574/wjarr.2026.30.3.1629,

46. **Naveen Undrathi,** Agentic generative AI-enabled microservices framework for intelligent financial data processing in java-based enterprise systems, September 2026 (Volume 31, Issue 3), eISSN: 2581-9615 || CODEN: WJARAI || Impact Factor 8.2 || https://doi.org/10.30574/wjarr.2026.30.3.1628

47. **Akshay Jain,** Comparative Analysis of Linkage Methods in Hierarchical Clustering for Customer Segmentation, Conference paper, First Online: 07 October 2025, https://link.springer.com/chapter/10.1007/978-981-96-7289-9_44

48. **Ismail Hossain,** Department of Computer Science, George Mason University, Fairfax, VA, USA, Mridul Banik, Department of Computer Science, Colorado State University, Fort Collins, CO, USA, https://ieeexplore.ieee.org/abstract/document/11446534 ,,DOI: 10.1109/FMLDS67896.2025.00106

49. **Ismail Hossain, Mridul Banik , Fahmid Al Farid , Jia Uddin , Hezerul bin Abdul Karim** , Advancing Large Language Models for Low-Resource Languages: A Systematic Review of Pretraining, Adaptation, and Ethical Challenges, 10.32604/cmes.2026.075507, Date: 2026 https://www.sciencedirect.com/org/science/article/pii/S152614922600322X

50. **Ismail Hossain; Mridul Banik,** Bridging Visual and Linguistic Intelligence for Chest X-rays: A Comprehensive Review of ViTs and LLM Synergies, Published in: 2025 IEEE International Conference on Future Machine Learning and Data Science (FMLDS) Date of Conference: 02-05 November 2025, Date Added to IEEE Xplore: 25 March 2026, ISBN Information: https://ieeexplore.ieee.org/abstract/document/11446534

51. **Rifat Sarker Aoyon1, Ismail Hossain2, M. Abdullah-Al-Wadud3, Jia Uddin4,** A Secured and Continuously Developing Methodology for Breast Cancer Image Segmentation via U-Net Based Architecture and Distributed Data Training, https://www.sciopen.com/article/10.32604/cmes.2025.060917

52. **M. B. Vibha, M. Chandrika, Samitha Khaiyum & P. Rakshitha Kiran,** Predicting sedentary behavior in adults using stacked LSTM modeling, published: 27 November 2024 Volume 16, pages 346–355 (2025), https://link.springer.com/article/10.1007/s13198-024-02622-2

53. **Bellary Chiterki Anil, Jayasimha Sondekoppa Rajkumar, T. L. Divya, Samitha khaiyum**

54. **Rakshitha Kiran P., Balakrishnan Ramadoss,** A Radiomics-based Framework for Liver Cancer Analysis using Explainable Artificial Intelligence (XAI) Methods, Vol. 15 No. 3 (2025): June, 2025, https://etasr.com/index.php/ETASR/article/view/10377

55. **Jayminkumar Patel, Sanjaykumar patel, Birju Patel, Rajan shah,** Development and Validation of a Stability-Indicating Analytical Method for Quantitative Determination of Selected Anticancer Drugs in Pharmaceutical Dosage Form, Archives / Vol. 35 No. S2 (2026): Apr-jun 2026, https://welltestingjournal.com/index.php/WT/article/view/288

56. **Jayminkumar Patel, Sanjaykumar patel, Birju Patel, Rajan shah**, Artificial Intelligence-Guided Optimization of Chromatographic Methods for Pharmaceutical Analytical Development, Archives / Vol. 35 No. S2 (2026): Apr-jun 2026, https://welltestingjournal.com/index.php/WT/article/view/294

57. **Sharoon Sharif, Inikpi Egbunu, Yujian Fu, and Zhigang Xiao. 2026**. Hybrid Multi-Modal Learning for Extreme Long-Tail Fungi Classification. In 2026 ACM Southeast Conference (ACMSE 2026), April 23–25, 2026, Troy, AL, USA. ACM, New York, NY, USA, 38–45. https://doi.org/10.1145/3746467.3801502

58. **A. Chandrakumaran**, "Innovative Strategies for Real-Time Analytics in Modern Data Warehousing Using Dockerized Ecosystems," 2025 International Conference on Advances in Next-Gen Computer Science (ICANCS), Bangalore, India, 2025, pp. 1-6, doi: https://doi.org/10.1109/ICANCS65819.2025.11377462

59. **A. Chandrakumaran,** "Enhancing SAP Predictive Maintenance and Industrial Analytics with LLM-Driven Automation," 2025 IEEE Madhya Pradesh Section Conference (MPCON), Jabalpur, India, 2025, pp. 238-244, doi: https://doi.org/10.1109/MPCON66082.2025.11256577.

60. **A. Chandrakumaran,** "Enhancing SAP Cloud Optimization with Explainable AI: Cost Control, Workload Forecasting, and Performance Enhancement," 2025 3rd International Conference on Inventive Computing

and Informatics (ICICI), Bangalore, India, 2025, pp. 456-463, doi: https://doi.org/10.1109/ICICI65870.2025.11069968.

61. **A. Chandrakumaran**, "Innovative Strategies for Real-Time Analytics in Modern Data Warehousing Using Dockerized Ecosystems," 2025 International Conference on Advances in Next-Gen Computer Science (ICANCS), Bangalore, India, 2025, pp. 1-6, doi: https://doi.org/10.1109/ICANCS65819.2025.11377462.

62. **Ashwini Chandrakumara**, year :2025, Leveraging Autonomous AI for Agile Growth: Transforming Project Management in Fast-Scaling Startups, Proceedings of the 2nd International Conference on Sustainable Business Practices and Innovative Models (ICSBPIM-2025), pages :730-743, issn :2352-5428, isbn :978-94-6463-872-1, Atlantis Press https://doi.org/10.2991/978-94-6463-872-1_46

63. **Anasuya Patil 1, Gurinderdeep Singh 2, Rajendra Dnyandeo Dighe 3, Dhruv Dev 4, Bhaveshkumar A Patel 5, Samatha Rudrangi 6, Gaurav Tiwari 7,** Preparation, optimization, and evaluation of ligand-tethered atovaquone-proguanil-loaded nanoparticles for malaria treatment, **2025 Apr;36(6):**711-742. doi: 10.1080/09205063.2024.2422704. Epub 2024 Nov 10., https://pubmed.ncbi.nlm.nih.gov/39522102/

64. **Kancharla Vijayalakshmi, Bethapudi Samuel Anand Andrews,** Estimation of Cyproheptadine Hydrochloride and Tricholine Citrate Simultaneously in Syrup Dose by Applying Stability Indicating HPLC Methodology, **December 31, 2020,** Vol. 11, Issue 4, Pages 1813–1819, DOI: 10.26452/ijrps.v11iSPL4.4383, https://ijrps.com/v11si4/3683

65. **Vijayalakshmi Kancharla ,Kumar Abbaraju V D N, Andrews B S A, Nagaraju Dasari, Nagu Korupolu,** Melphan as Anticancer and New Insights of Drug as Parenteral Dosage form against SARS-CoV-2 by Using Development and Validation , **November 18, 2020**, Vol. 11, Issue 1, Pages 1437–1446 DOI: 10.26452/ijrps.v11iSPL1.3686, https://ijrps.com/v11si1/2911

66. **Ramesh, D., A. Singh, and L. H. Channaiah**. 2026. " Evaluating the Effectiveness of Chewy Chocolate Chip Granola Bar Manufacturing Process on Salmonella Survival." Food Science & Nutrition 14, no. 4: e71698. https://doi.org/10.1002/fsn3.71698.

67. **Inventor: Akhilesh RaiDilip Srinivas Krovi Sarthak Saptarshi Bandopadhyay Ravinder Singh Choudhary Richard LAWTON, 2025-02-28,** Priority to US19/067,141,2025-11-14,Assigned to CITIBANK, N.A.,2025-12-30,Application granted,2025-12-30, Publication of US12511187B, https://patents.google.com/patent/US12511187B1/en?oq=US12511187

68. **Vivekanandan Govindan Ekambaram ,** REAL-TIME AUTOMATED DEFECT DETECTION IN INDUSTRIAL MANUFACTURING USING DEEP LEARNING AND TEXTURE ANALYSIS, Archives ,Vol. 54 No. 4 (2026): **October-December 2026**, Power System Protection and Control, ISSN-1674-3415,DOI: https://doi.org/10.46121/pspc.54.4.06, https://pspac.info/index.php/dlbh/article/view/545

69. **Vivekanandan Govindan Ekambaram, Yamuna Natarajan,** AN OPTIMIZED HYBRID FRAMEWORK FOR REAL-TIME EDGE DETECTION AND FEATURE EXTRACTION IN AUTONOMOUS VEHICLE VISION

SYSTEMS, **Vol. 54 No. 4 (2026):** October-December 2026 , Power System Protection and Control, ISSN-1674-3415, DOI: https://doi.org/10.46121/pspc.54.4.05, https://pspac.info/index.php/dlbh/article/view/544

70. **John Smart Mgbakor,** Behavioral Health Equity In High-Disparity Urban Communities: Public Health Impact, National Scalability, And Clinical Practice Implications, Pages1166 – 1177Cover Date, **August 2026,** Volume34, Issue, Special Issue 7- Fundamental Studies, Print ISSN-0965-7452, Electronic-ISSN-2396-8893, DOI: 10.1922/ejprd.v34i7s.1767, https://www.ejprd.org/article.php?article_id=1767&journal_id=171

71. **1: Nikhil Kumar Singh, Binay K. Singh, Piyush Kumar, Avinash Pandey, Sudhir Kumar, Sujit Kumar Bishi, A. Pattanayak, V. P. Bhadana, Sujay Rakshit & Kishor U. Tribhuvan,** Integrative genomic and machine learning approaches reveal evolutionary signatures in the winged bean mitochondrial genome, Published: **28 November 2025,**Volume 26, article number 3 (2026), https://link.springer.com/article/10.1186/s12870-025-07738-6

72. GAURAV KUMAR SINHA, Year 2026 https://mathsapplication.com/article-view/?id=223

73. GAURAV KUMAR SINHA, Year 2026 , https://mathsapplication.com/article-view/?id=224

74. GAURAV KUMAR SINHA, Year 2026 , https://mathsapplication.com/article-view/?id=225

75. Rasmi Ranjan Choudhury, Design and Implementation of a Tuned Rear Axle Bending Damper for Beat Boom Noise Mitigation in SUV Driveline Systems, **2026 International Conference, https://ieeexplore.ieee.org/document/11746870/authors#authors**

76. Rasmi Ranjan Choudhury, Comparative Performance Analysis of Single-Speed Versus Multi-Speed Transmissions in Battery Electric, **2026 International Conference, https://ieeexplore.ieee.org/document/11746704**

77. **Nithesh Gudipuri1, Lalith Chandra Bandaru2, Mohammed Shakeer Bandrevu,** SCALABLE DATA ENGINEERING FRAMEWORKS FOR REAL-TIME PROCESSING IN DISTRIBUTED SYSTEMS, Published:**2026-10-18**,Pages:444-453, https://actainnovations.com/index.php/pub/article/view/1900 ,doi: 10.62441/actainnovations.vi.1900

78. **Mohammed Shakeer Bandrevu; Pratham Patkar; Janardhana Naidu Kola,** Trustworthy AI Models for Secure Data Processing Environments, **Published in: 2026 International Conference on Natural Language Processing and Computer Vision (ICNPCV),Date** of Conference: 17-18 July 2026, DOI: 10.1109/ICNPCV69434.2026.11708997, https://ieeexplore.ieee.org/document/11708997/authors#authors,

79. **Lalith Chandra Bandaru; Venkateswara Rao Batta; Uday Kumar Reddy Gangula**, Hybrid AI and Data Analytics Frameworks for Cyber Resilience**, 2026 International** Conference on Natural Language Processing and Computer Vision (ICNPCV),Date of Conference: 17-18 July 2026,DOI:

10.1109/ICNPCV69434.2026.11708795, https://ieeexplore.ieee.org/document/11708795

80. **Lalith Chandra Bandaru; Karimulla Syed,Head of Access Management, S&P Global, Raleigh, NC, USA; Nishanth Vepachedu; Rajmani Singh Oinam; Raja Vignesh Vaithiyalingam; Quazi Taif Sadat,** Advanced Secure Knowledge Extraction Techniques for Large Language Models, **2026 International** Conference on Emerging in Computing Technologies (INCECT),Date of Conference: 21-22 **August 2026, DOI: 10.1109/INCECT72370.2026.11742788, https://ieeexplore.ieee.org/document/11742788**

81. **Mohammed Shakeer Bandrevu, Independent Researcher, Aubrey, TX, USA, ; NagaSatyanarayana Raju Uppalapati; Ramesh Bellamkonda; Srikanth Kamatala**, Llm-Augmented Cyber Threat Detection Using Federated Edge Intelligence and Data Mining Techniques, **Published in: 2025** International Conference on Computing Technologies & Data Communication (ICCTDC),Date of Conference: 04-05 July 2025, https://ieeexplore.ieee.org/document/11158153 , DOI: 10.1109/ICCTDC64446.2025.11158153
82. **Ankit Pathak**, Leveraging AI for Better Data Quality and Insights, **Volume 7 Issue 3 (2025),** https://doi.org/10.32996/jcsts.2025.7.2.19, https://al-kindipublisher.com/index.php/jcsts/article/view/9345

83. **Uma Rajeswari Batchu, Bharati Reddi, Joshna Rani Surapaneni, Aruna Jangam, Sunil Misra, Anthony Addlagatta & Prakasham Reddy Shetty,** Inhibition of Xanthine Oxidase by 1-O-Methyl Chrysophanol, A Hydroxyanthraquinone Isolated from Amycolatopsis Thermoflava ICTA 103, Published: **13 December 2025,**Volume 66, pages 1049–1068 (2026) https://link.springer.com/article/10.1007/s12088-025-01534-4

84. GAURAV KUMAR SINHA, Year 2025, https://visualcompublications.es/SAUC/article/view/6525

85. GAURAV KUMAR SINHA, Year 2025, https://visualcompublications.es/SAUC/article/view/6526

86. GAURAV KUMAR SINHA, Year 2025, https://visualcompublications.es/SAUC/article/view/6527

87. **D Benito Johnson 1, Gurinderdeep Singh 2, Deeksha Sharma 3, Venkatesan Natarajan 4, Knv Chenchu Lakshmi 5, Ram C Dhakar 6, Sadhana R Shahi 7, Suresh Velayutham 8, Ruchi Tiwari 9,** Exploring Computational Advancements in ADME: Essential Insights for Drug Disposition, PMID: 39496312 DOI: 10.62958/j.cjap.**2024.**033, https://pubmed.ncbi.nlm.nih.gov/39496312/

88. Mohammed Shakeer Bandrevu, Service Now Ebonding Methods, Implementation and Key features Published,**30 Nov 2024**, https://ijcttjournal.org/archives/ijctt-v72i11p125