A Decoy Placement Model for Engineering Workstations and Historian Systems in Manufacturing OT Networks

Daniel Ward
Southern New Hampshire University
Manchester, New Hampshire, United States

ABSTRACT

Engineering workstations and historian systems concentrate control-project artifacts, privileged maintenance paths, and process knowledge in manufacturing operational technology (OT) networks. This paper develops and evaluates a safety-constrained decoy placement model for those assets. The revised study combines design science with a reproducible computational experiment built on a 26-node, 54-edge directed manufacturing OT attack graph. Six attack scenarios were tested against a no-deception baseline and four equal-budget alternatives: demilitarized-zone-heavy, random, betweenness-centrality, and asset-proximal placement. Each strategy was evaluated with 10,000 Monte Carlo trials per scenario under a five-decoy budget, resulting in 360,000 nominal trials with an arbitrary fixed seed of 314159. The proposed distributed placement achieved weighted pre-impact detection of 0.639 (95% CI 0.636-0.643), compared with 0.587 for asset-proximal placement, 0.574 for centrality placement, 0.411 for random placement, 0.356 for DMZ-heavy placement, and 0.122 for the baseline. Critical-target reach declined to 0.361. The advantage remained under reduced-fidelity and attacker-policy sensitivity tests. The results support distributed, high-value placement across remote access, engineering, project-file, historian, and protocol-discovery paths. The experiment is synthetic and does not claim live-plant effectiveness, but it provides falsifiable, reproducible evidence for comparative placement decisions.

General Terms

Security, Algorithms, Design, Experimentation, Measurement

Keywords

Cyber deception, decoy placement, engineering workstations, historian systems, industrial control systems, operational technology, Monte Carlo evaluation

1. INTRODUCTION

Manufacturing OT networks connect enterprise planning, vendor support, plant operations, engineering maintenance, industrial data services, and production equipment. Engineering workstations and historian systems occupy especially important positions in this structure. Engineering workstations may contain controller projects, vendor programming environments, firmware, configuration files, network drawings, certificates, and remote connection profiles. Historians aggregate tag names, process values, alarms, batch context, quality variables, and reporting interfaces. Unauthorized access to either asset can support discovery, credential misuse, intellectual-property collection, lateral movement, or preparation for process disruption [1], [2].

Deception technology can complement firewalls, endpoint protection, asset inventory, segmentation, and monitoring by presenting resources that legitimate users and systems should not touch. Interaction with a decoy engineering workstation, a synthetic project share, a historian endpoint, a honey credential, or a protocol emulator can therefore create a high-confidence signal. The deployment problem is placement. A decoy may be technically functional yet strategically weak if it is concentrated in one zone, disconnected from likely attack paths, too easy to fingerprint, or positioned where legitimate activity creates noise. In OT, poor placement can also create operational concern if a decoy resembles a controller, advertises unrealistic process behavior, or introduces an unintended management path [2], [8]-[12].

This paper comprises a controlled computational experiment and a comparative analysis. The study does not fabricate live-plant results. Instead, it makes the topology, attack scenarios, parameters, distributions, seed, and code explicit so that the placement claims can be reproduced and challenged.

The contributions are fourfold. First, the paper defines a five-class placement model centered on engineering workstations and historian systems. Second, it creates a transparent manufacturing OT attack graph and six attack scenarios. Third, it compares the proposed distribution against five alternatives under the same decoy budget. Fourth, it reports confidence intervals, scenario-level results, operational-burden proxies, and sensitivity analysis.

2. BACKGROUND AND RESEARCH QUESTIONS

2.1 Engineering Workstations and Historians

Engineering workstations represent a control authority. They are used to create, download, back up, troubleshoot, and validate automation logic. Historians represent process knowledge. They collect data that can reveal operating ranges, production schedules, equipment relationships, and data-flow dependencies. These functions are attractive to adversaries but also lend themselves to safe imitation because decoys can expose synthetic directories, hostnames, tags, connection strings, and read-only services without copying real controller logic or production data.

2.2 Deception as High-Confidence Telemetry

Honeypot and honeynet research across IoT, IIoT, cyber-physical systems, and ICS emphasizes realism, observability, containment, and attacker engagement [7]. ICS-specific studies show the progression from low-interaction emulation toward high-interaction and physics-aware designs [8]-[11]. HoneyPLC demonstrated that a PLC-oriented honeypot can imitate vendor-specific behavior and engage reconnaissance tools [9]. HoneyICS and ICSNet further show that industrial decoys can incorporate process-aware or hybrid-interaction behavior [10], [11]. These studies also reinforce a central limitation: fidelity and safety must be balanced. If a decoy is obviously artificial, an adversary may avoid it; if it is too deeply connected, it may create unnecessary risk [12].

Prior control-mapping and metrics work argued that OT deception should be governed, isolated, linked to response playbooks, and evaluated through alert quality rather than event volume alone [13], [14]. The present study narrows that

argument to the issue of placement. It asks whether a distributed set of decoys around remote access, engineering, project-file, historian, and protocol-discovery paths can intercept more attack paths than common placement alternatives.

2.3 Evaluation Questions

RQ1. Under an equal five-decoy budget, does the proposed placement model increase pre-impact detection and reduce critical-target reach relative to no-deception, DMZ-heavy, random, centrality-based, and asset-proximal placement?

RQ2. How consistently does each strategy perform across enterprise-to-OT movement, stolen vendor credentials, engineering project collection, historian collection, cell/area protocol discovery, and compromised OT identity scenarios?

RQ3. Does the proposed model retain its comparative advantage when decoy fidelity and attacker goal-directedness are varied?

RQ4. What operational burden is implied by each strategy under a transparent synthetic benign-interaction proxy?

3. RESEARCH DESIGN AND METHODS

3.1 Design Science Artifact and Decoy Classes

The study uses design science followed by controlled computational evaluation. No human participants, private organizational records, live industrial traffic, or production control systems were used. The artifact translates OT security requirements, deception research, and prior adoption findings into five decoy classes. Table 1 summarizes their intended locations, signals, and safety boundaries.

Table 1. Core decoy classes in the proposed placement model

Decoy class	Primary placement	Primary detection signal	Safety boundary
Vendor-access lure	Industrial DMZ or vendor landing zone	Authentication, SSH/RDP/VPN correlation, source identity	No valid route or credential to production
Engineering-workstation decoy	Controlled maintenance subnet or engineering enclave	Logon, service scan, file browsing, remote-desktop probe	Synthetic project content; no controller trust
Synthetic project share	Decoy workstation, backup path, or controlled file share	Open/copy event, account, hash, source, timestamp	No real logic, passwords, or intellectual property
Historian-endpoint decoy	Historian subnet, analytics staging segment, or reporting enclave	Query, tag enumeration, API/ODBC attempt, source identity	Synthetic values; no bridge to production historian
Read-only protocol emulator	Isolated cell/area deception subnet	Industrial discovery, read request, scan pattern, packet capture	No write functions and no route to live controllers

3.2 Comparator Strategies

All active strategies used five decoys so that differences reflected placement rather than budget. The centrality strategy selected one candidate per mimicked asset and then chose the five highest-betweenness candidate locations in the base graph. The random strategy resampled five of the fourteen candidates without replacement for every trial. Table 2 defines the complete comparison.

Table 2. Equal-budget placement strategies

Strategy	Portfolio or rule	Evaluation rationale
No-deception baseline	No decoys	Measures conventional detection assumptions only
DMZ-heavy	D1, D2, D3, D4, D13	Concentrates lures around remote access, patching, file transfer, and historian relay
Random placement	Five of fourteen candidates resampled per trial	Estimates expected performance without placement guidance
Centrality-based	D1, D5, D10, D9, D7	Selects high-betweenness decoy-eligible locations
Asset-proximal	D5, D7, D9, D10, D11	Clusters decoys near engineering, historian, SCADA, OPC, and controller-adjacent assets
Proposed model	D1, D5, D6, D7, D11	Distributes decoys across remote access, engineering, project, historian, and protocol paths

3.3 Synthetic Manufacturing OT Attack Graph

The experiment uses a directed graph containing 26 nodes and 54 edges. Nodes represent external access, enterprise endpoints, and identity services, an industrial DMZ, site OT operations, cell/area infrastructure, and production assets. Edges represent feasible discovery or lateral-movement transitions, not packet-level network reachability. The graph includes vendor access, jump hosts, patch and file-transfer services, historian relays, OT identity, engineering workstations, project shares, historian services, SCADA/HMI, OPC services, cell switching, PLCs, robots, and drives. Fourteen decoy candidates are attached as plausible alternatives to selected real nodes.

Six scenarios were selected because they exercise the high-value paths addressed by the placement model. Scenario weights sum to one and are used only to produce an aggregate decision metric; all scenario-specific results are also reported separately.

Table 3. Synthetic attack scenarios

ID	Scenario	Starting position	Critical target set	Weight
S1	Enterprise-to-OT lateral movement	Employee workstation	Engineering, historian, PLC, or robot	0.20
S2	Stolen vendor credentials	VPN service	Engineering, SCADA, or PLC	0.20
S3	Engineering project collection	DMZ jump host	Engineering workstation or project share	0.15
S4	Historian reconnaissance and collection	Enterprise application server	Production historian	0.15
S5	Cell/area protocol discovery	SCADA node	PLC, robot, or drive	0.15
S6	Compromised OT identity or insider	OT identity service	Engineering, historian, SCADA, or PLC	0.15

3.4 Attacker Transition and Detection Model

At each step, the attacker selects a reachable real node or a visible decoy. A real next hop receives a weight proportional to exp[-alpha d(u,T)] with a small zone-depth adjustment, where d(u,T) is the shortest directed distance from candidate node u to the scenario target set T. A visible decoy receives a weight based on the distance of the asset it imitates, a scenario-specific affinity, and its fidelity. Weights are normalized into next-hop probabilities. The nominal goal-directedness parameter alpha is 0.95; lower values indicate more exploratory movement, and higher values indicate more direct movement.

When a decoy is selected, the attacker interacts with probability equal to the candidate-specific fidelity. A successful interaction generates an alert with the candidate-specific alert-reliability probability. A rejected or non-alerting decoy is blacklisted for the remainder of that trial. Real transitions retain the same zone-specific conventional-detection probability across all strategies. A trial ends when a decoy or conventional control detects the activity, a critical target is reached, no path remains, or fourteen steps are completed.

3.5 Parameters, Replication, and Statistical Analysis

Table 4 lists the complete nominal and sensitivity assumptions used in the computational experiment, including the graph size, decoy budget, replication counts, fixed arbitrary random seed, path-length limit, attacker-policy values, candidate-fidelity range, alert reliability, and conventional-detection probabilities.

Table 4. Computational experiment assumptions

Parameter	Nominal value	Purpose
Graph size	26 nodes; 54 directed edges	Synthetic manufacturing architecture
Decoy candidates / budget	14 candidates; 5 active decoys	Equal-budget comparison
Nominal replication	10,000 trials per scenario per strategy	360,000 nominal attack trials
Sensitivity replication	2,000 trials per scenario, strategy, and condition	420,000 additional trials
Random seed	314159	Arbitrary non-date seed for NumPy PCG64 reproducibility
Maximum path length	14 transitions	Prevents indefinite cycling
Goal-directedness	alpha = 0.95 nominal; 0.60 and 1.30 sensitivity	Exploratory to direct attacker policy
Decoy fidelity	0.72-0.94 by candidate; multipliers 0.70-1.10	Tests fingerprinting/realism sensitivity
Alert reliability	0.975-0.995 by candidate	Models telemetry delivery after interaction
Conventional detection	0.035 enterprise to 0.075 production	Held constant across all strategies

Note. All values are analytical assumptions for comparative evaluation. They are not estimates of universal attacker behavior or industrial incident rates.

The primary outcomes are pre-impact detection probability, decoy-specific detection probability, critical-target reach probability, conditional mean steps to detection, and the number of scenarios with detection at or above 0.50. Scenario-level proportions use 95% Wilson intervals. Weighted aggregate intervals use the stratified binomial standard error. Pairwise differences are reported with 95% confidence intervals. A synthetic benign-alert proxy sums candidate-specific accidental-interaction rates per 10,000 legitimate sessions; it is an operational-burden assumption, not a measured field false-positive rate. NumPy and NetworkX were used for pseudo-random simulation and graph analysis [15], [16]. The seed is an arbitrary numerical constant, selected solely for deterministic reproducibility, and does not encode the analysis date.

4. DECOY PLACEMENT MODEL

4.1 Architecture and Noninterference

Figure 1 places the vendor-access lure in the industrial DMZ, the engineering workstation, project share, and historian decoys in site OT operations, and the read-only protocol emulator in an isolated cell/area deception subnet. Telemetry is exported to the enterprise security operations function. The architecture intentionally omits a command or management route from the decoys into production assets.

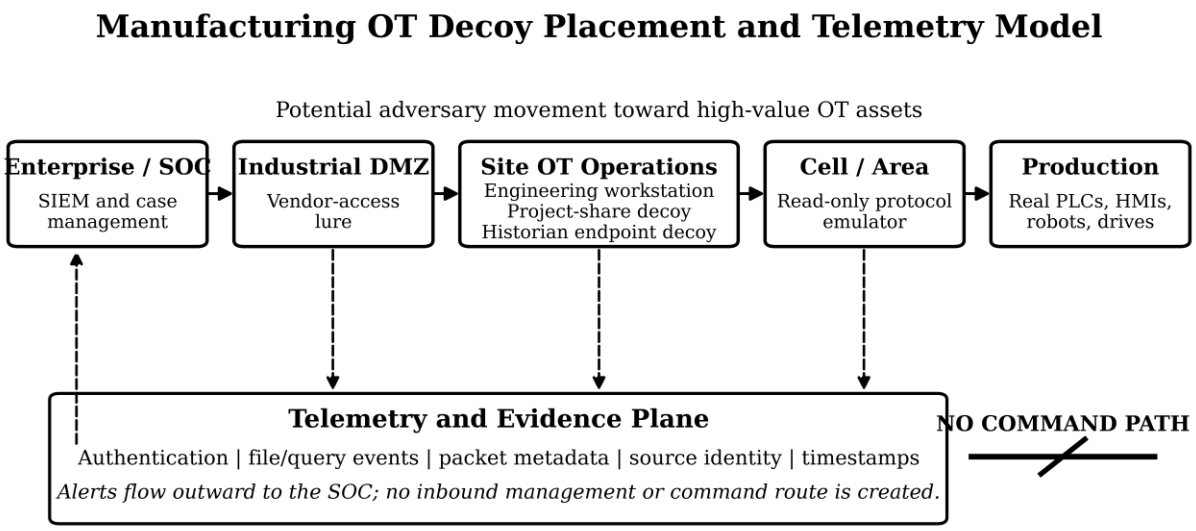

Figure 1. Manufacturing OT decoy placement and telemetry architecture. Solid arrows show possible adversary movement; dashed arrows show telemetry flow. The proposed placement spans distinct trust boundaries while preserving a no-command-path rule.

4.2 Placement Logic

The model uses distributed semantic coverage rather than concentration in a single network zone. D1 observes vendor and remote-access pathways before an attacker reaches the site OT. D5 and D6 imitate the engineering authority and project artifacts that support controller maintenance. D7 presents a synthetic historian target for process-data reconnaissance. D11 observes industrial protocol discovery near the cell/area boundary but does not provide control authority. This combination was selected because it covers different attacker goals and different points in the attack path, not because the five decoys have the highest individual centrality scores.

The safety logic is equally important. The project and historian artifacts are synthetic; the protocol emulator is read-only and isolated; no decoy is part of the operator workflow; and alert telemetry moves outward. These constraints reduce the likelihood that the detection layer becomes a production dependency or an unreviewed route to controllers.

5. EXPERIMENTAL RESULTS

5.1 Aggregate Comparative Performance

Figure 2 and Table 5 answer RQ1. The proposed model achieved weighted pre-impact detection of 0.639 (95% CI 0.636-0.643). This was 0.052 higher than asset-proximal placement (95% CI for the difference 0.047-0.058), 0.065 higher than centrality-based placement (0.060-0.071), 0.228 higher than random placement (0.223-0.234), and 0.284 higher than DMZ-heavy placement (0.278-0.289). Relative to the no-deception baseline, the absolute increase was 0.518.

The critical-target reach result moves in the opposite direction. The proposed strategy reduced target reach to 0.361, compared with 0.413 for asset-proximal, 0.426 for centrality-based, 0.589 for random, 0.644 for DMZ-heavy, and 0.878 for the baseline. Decoys accounted for 0.574 of the proposed model's detections, indicating that the gain was not due to changes in the conventional monitoring assumptions.

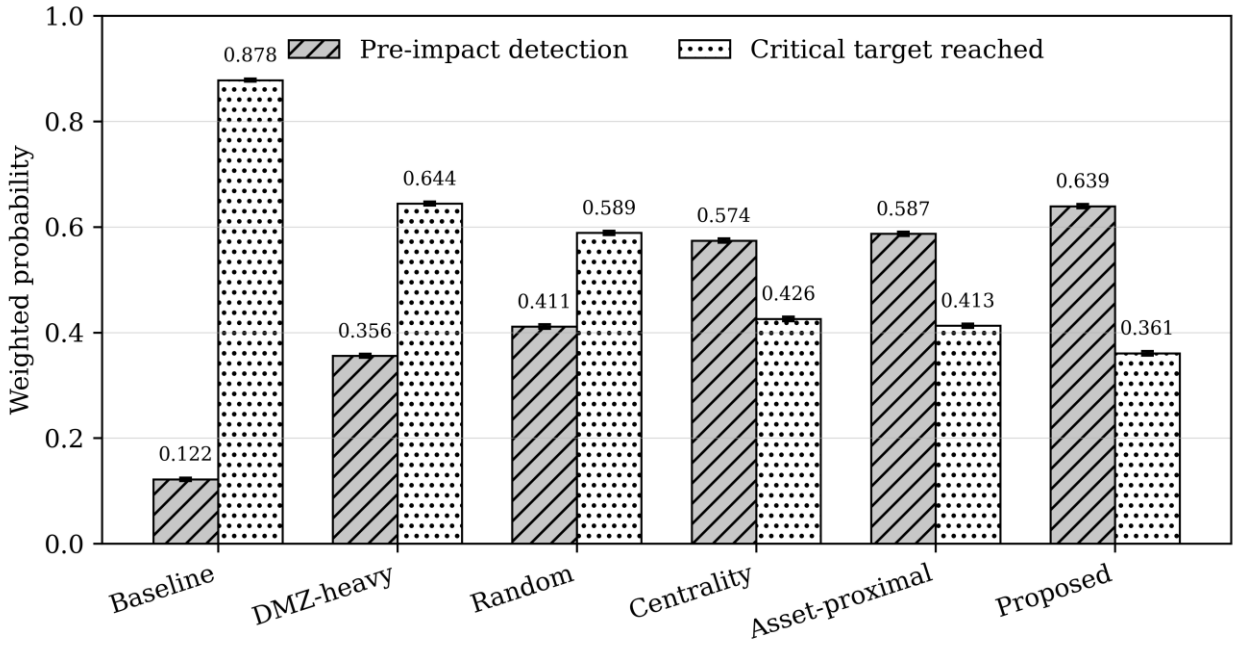

Figure 2. Weighted pre-impact detection and critical-target reach across equal-budget placement strategies. Error bars show 95% confidence intervals; the intervals are narrow because each scenario-strategy cell contains 10,000 trials.

Table 5. Aggregate placement-strategy results

Strategy	Pre-impact detection [95% CI]	Target reach	Decoy detection	Mean steps	Coverage	Benign-alert proxy
No-deception baseline	0.122 [0.119, 0.124]	0.878	0.000	1.88	0/6	0.0
DMZ-heavy	0.356 [0.352, 0.359]	0.644	0.265	1.51	1/6	18.8
Random placement	0.411 [0.407, 0.415]	0.589	0.324	1.73	1/6	13.6
Centrality-based	0.574 [0.570, 0.578]	0.426	0.501	1.79	4/6	9.5
Asset-proximal	0.587 [0.583, 0.591]	0.413	0.511	1.98	5/6	8.8
Proposed model	0.639 [0.636, 0.643]	0.361	0.574	1.80	5/6	3.2

Note. Probabilities are scenario-weighted. Mean steps are conditional on detection. The benign-alert value is a synthetic proxy per 10,000 legitimate sessions, not a field false-positive estimate.

5.2 Scenario-Level Performance

Figure 3 and Table 6 answer RQ2. The DMZ-heavy strategy performed strongly for stolen vendor credentials (0.757) but detected only 0.143 of cell/area protocol discovery trials and 0.062 of compromised-OT-identity trials. This illustrates the weakness of zone concentration. Centrality placement performed best in the vendor-credential scenario (0.784), but it lacked the dedicated protocol emulator and fell to 0.360 for cell/area discovery. Asset-proximal placement produced the highest cell/area result (0.777), but its lack of a remote-access lure reduced vendor-scenario detection to 0.539.

The proposed model did not win every scenario. It was second to centrality placement for stolen vendor credentials and second to asset-proximal placement for cell/area discovery. Its advantage was balance: 0.616 for enterprise-to-OT movement, 0.769 for vendor credentials, 0.577 for project collection, 0.654 for historian collection, 0.720 for protocol discovery, and 0.466 for compromised OT identity. Five of six scenarios exceeded 0.50, and the sixth was higher than every comparator. The result supports a distributed design that covers distinct adversary objectives rather than optimizing a single path.

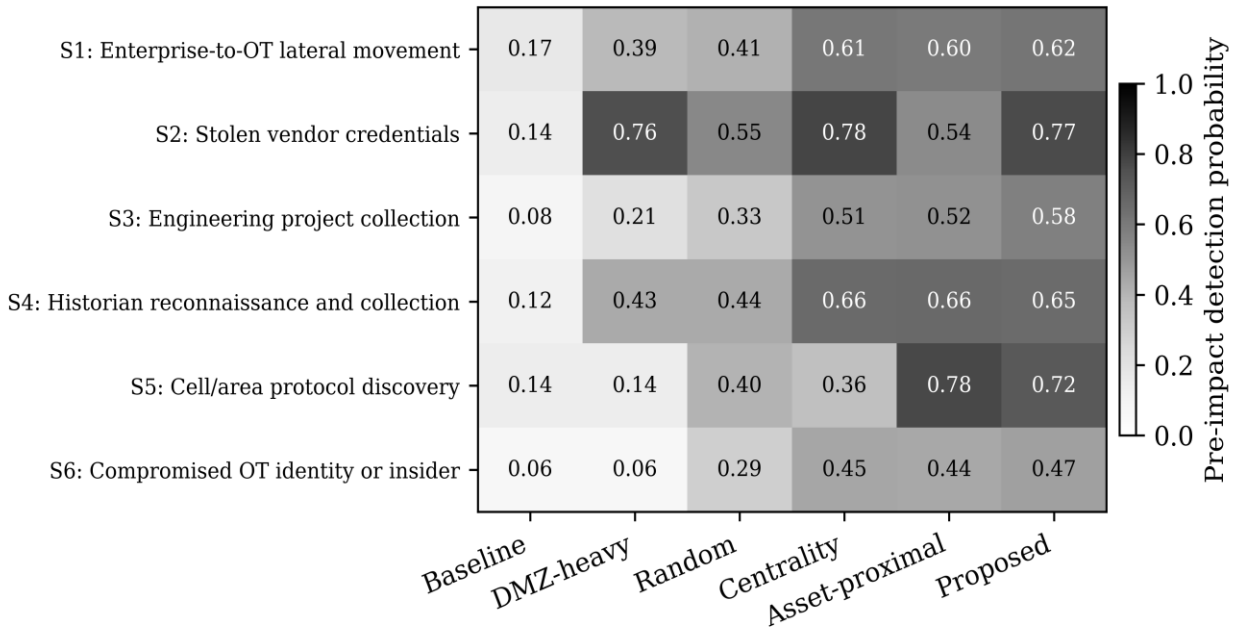

Figure 3. Scenario-level pre-impact detection probabilities. Darker cells indicate stronger interception. The proposed placement provides the most balanced coverage across remote access, engineering, historian, and protocol-discovery scenarios.

Table 6. Scenario-level pre-impact detection probabilities

Strategy	S1	S2	S3	S4	S5	S6
No-deception baseline	0.165	0.143	0.080	0.117	0.143	0.063
DMZ-heavy	0.386	0.757	0.209	0.434	0.143	0.062
Random placement	0.413	0.549	0.326	0.436	0.405	0.293
Centrality-based	0.606	0.784	0.510	0.659	0.360	0.446
Asset-proximal	0.600	0.539	0.517	0.661	0.777	0.441
Proposed model	0.616	0.769	0.577	0.654	0.720	0.466

Note. Each cell is based on 10,000 trials. Scenario definitions appear in Table 3.

5.3 Operational Burden Proxy

The proposed model also produced the lowest synthetic benign-alert burden among active strategies: 3.2 alerts per 10,000 legitimate sessions, compared with 8.8 for asset-proximal, 9.5 for centrality-based, 13.6 for random, and 18.8 for DMZ-heavy placement. The result follows from the placement assumptions. The proposed project-share, historian, and protocol decoys are positioned where legitimate users and automated systems are expected to have zero or near-zero interaction. DMZ patch and file-transfer lures are more exposed to benign administrative activity and misconfiguration. This finding should be interpreted as a design comparison, not as a prediction of actual plant false positives.

5.4 Sensitivity and Robustness

Figure 4 and Table 7 answer RQ3. When every candidate's fidelity was multiplied by 0.70, the proposed strategy still produced the highest detection probability (0.478), ahead of the asset-proximal (0.441) and centrality-based (0.427) strategies. At the nominal multiplier, the sensitivity rerun produced 0.647 for the proposed model, and at 1.10, it reached 0.694. The ranking was also stable when attacker goal-directedness alpha varied from 0.60 to 1.30. Proposed-model detection remained between 0.639 and 0.647, while the nearest competitor remained below 0.585.

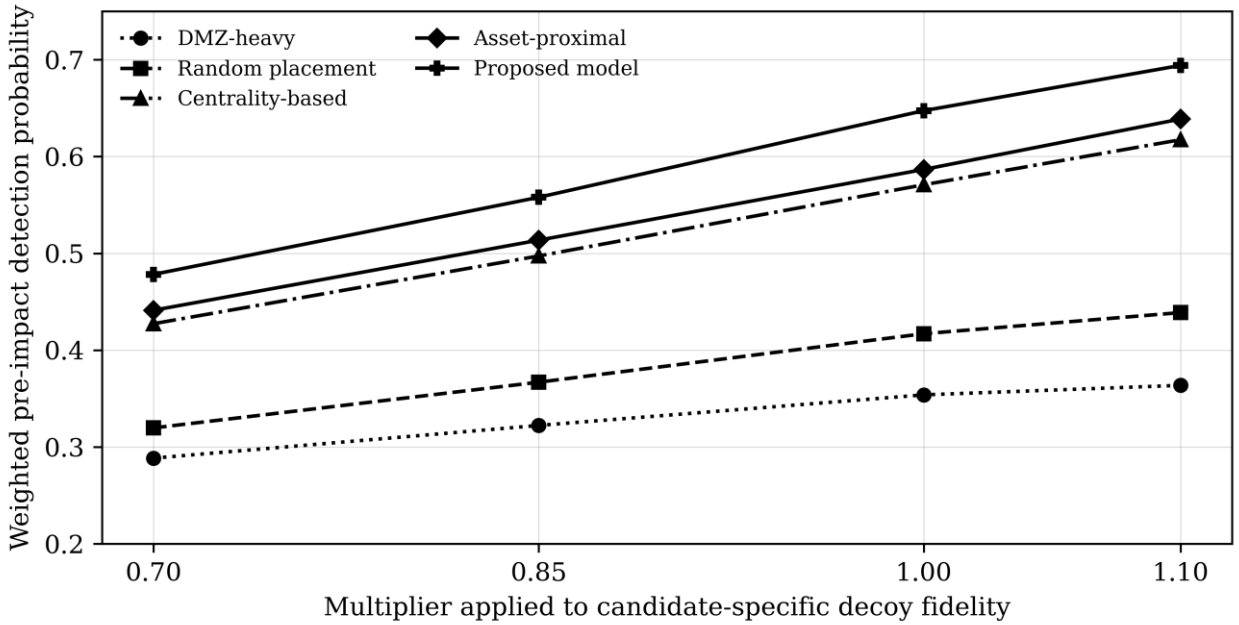

Figure 4. Fidelity sensitivity. The proposed model retains the highest weighted detection probability across low, nominal, and enhanced candidate-specific fidelity assumptions.

Table 7. Robustness checks for the three strongest placement strategies

Condition	Centrality	Asset-proximal	Proposed
Fidelity x 0.70	0.427	0.441	0.478
Fidelity x 0.85	0.497	0.514	0.558
Fidelity x 1.00	0.571	0.587	0.647
Fidelity x 1.10	0.617	0.639	0.694
Attacker alpha 0.60	0.573	0.585	0.639
Attacker alpha 0.95	0.579	0.584	0.646
Attacker alpha 1.30	0.568	0.582	0.647

Note. Sensitivity cells use 2,000 trials per scenario, strategy, and condition.

5.5 Interpretation of Robustness Tests

The fidelity analysis tests whether the proposed ranking depends on unrealistically effective decoys. It does not. Relative to asset-proximal placement, the proposed model retained absolute detection advantages of approximately 0.037, 0.044, 0.061, and 0.055 at fidelity multipliers of 0.70, 0.85, 1.00, and 1.10, respectively. The advantage over centrality-based placement ranged from approximately 0.051 to 0.077. Lower fidelity reduced the effectiveness of every strategy, but it did not erase the value of distributing decoys across remote-access, engineering, historian, and protocol-discovery paths.

The attacker-policy test addresses a different concern: whether the result is an artifact of an attacker wandering randomly through the graph. As goal-directedness increased from 0.60 to 1.30, the proposed model remained between 0.639 and 0.647 and stayed approximately 0.054-0.066 above the nearest comparator. Because conventional-detection probabilities were held constant across strategies, this stability is attributable to portfolio placement and route coverage within the model rather than to a more favorable monitoring baseline. These results do not establish field effectiveness, but they show that the comparative conclusion is not confined to one nominal parameter setting.

6. DISCUSSION

6.1 Why Distributed Placement Performed Better

The results show that decoy value is path-dependent. DMZ concentration captured remote-access behavior but provided little value after an adversary began inside site OT. Asset-proximal placement captured late-stage protocol discovery but surrendered early opportunities around vendor access and project artifacts. Centrality placement covered structurally important nodes, but graph centrality alone did not represent the semantic value of project shares or the special need for a controller-adjacent read-only emulator. The proposed model combined early, middle, and late interception points.

This distinction matters for engineering workstations and historians. These assets are not simply high-centrality servers. They embody different adversary objectives. Engineering decoys expose interest in control authority and project files. Historian decoys expose interest in process knowledge and data interfaces. A placement model that addresses both objectives can outperform one optimized only for network structure.

6.2 Practical Deployment Implications

A plant can implement the model incrementally. The first stage should establish the synthetic project share and the vendor-access

lure, as they are comparatively easy to isolate and test. The second stage should add engineering-workstation and historian decoys with explicit zero-legitimate-use statements, SIEM rules, owner approval, and response playbooks. The read-only protocol emulator should be deployed last, in an isolated deception subnet, after engineering review and noninterference testing. Every alert should record source, target, account, protocol, timestamp, disposition, engineering notification, and ATT&CK for ICS mapping where applicable [6], [13], [14].

The results do not imply that every manufacturer should deploy the same hostnames, protocols, or products. The model is vendor-neutral. Organizations should preserve the placement logic while adapting fidelity, naming conventions, synthetic content, and telemetry to their own architecture. The primary operational rule is unchanged: deception may observe and report, but it must not become part of process control.

6.3 Relationship to Existing Research

The study complements high-interaction ICS honeypot research rather than replacing it. HoneyPLC, HoneyICS, and ICSNet demonstrate how realistic industrial services can be built and evaluated [9]-[11]. The present contribution addresses where a limited number of those capabilities should be placed around engineering and historian functions. It also extends prior control-mapping and metrics work by adding a comparative placement experiment with explicit uncertainty and reproducibility [13], [14].

7. LIMITATIONS AND FUTURE WORK

The experiment is synthetic. The 26-node graph is representative, not a model of a named plant. The edge structure, scenario weights, fidelity values, alert reliability, conventional-detection probabilities, and benign-interaction rates are analytical assumptions. The attacker policy is stochastic and distance-aware but does not reproduce every credential, exploit, privilege, timing, or anti-honeypot behavior. The confidence intervals quantify Monte Carlo uncertainty under the model; they do not quantify uncertainty about whether the model matches a specific production environment.

The experiment also treats decoy interactions as terminal detections and does not model analyst queueing, containment delay, adaptive adversaries across multiple campaigns, correlated control failures, or safety consequences. The benign-alert metric is a proxy rather than observed false-positive behavior. The graph focuses on engineering workstations and historians; robotics, quality systems, safety instrumented systems, IIoT gateways, and building automation may require different candidate sets.

Future work should implement the same scenarios in a manufacturing cyber range using containerized services, a simulated historian, synthetic controller projects, controlled attack scripts, and SIEM telemetry. That validation should compare simulated and observed path selection, detection latency, false positives, fingerprinting, alert enrichment, and analyst response time. A second extension should calibrate graph edges and candidate fidelity from sector-specific architectures. A third should optimize placement under variable budgets and explicit operational-risk constraints rather than evaluating a fixed five-decoy artifact.

8. CONCLUSION

This study revised a qualitative decoy placement artifact into a reproducible comparative experiment. Under an equal five-decoy budget, the proposed distributed model achieved the highest weighted pre-impact detection probability and the lowest critical-target reach among the evaluated strategies. It also provided the most balanced scenario coverage and the lowest synthetic benign-alert burden. The model did not dominate every individual scenario, which is an important result: placement should balance remote access, engineering authority, project artifacts, historian knowledge, and protocol discovery rather than optimizing for a single pathway.

The practical contribution is a defensible placement rationale for manufacturing OT. A vendor-access lure, engineering-workstation decoy, synthetic project share, historian endpoint, and isolated read-only protocol emulator together create detection opportunities across multiple trust boundaries without creating a command path to production. The experiment remains synthetic, but its assumptions, seed, code, raw outcomes, graphical results, and tabular results are available for replication and future cyber-range validation.

9. DATA AND CODE AVAILABILITY

The manuscript submission includes the Python source code, a fixed-seed manifest, a directed graph edge list, candidate decoy definitions, attack-scenario definitions, strategy definitions, compressed nominal trial outcomes, summary CSV files, sensitivity results, and separate high-resolution figures. No private plant data or human-subject data are included.

10. ACKNOWLEDGMENTS

No external funding was received for this manuscript. The paper builds on the author's prior doctoral research and draws on public standards, public security frameworks, published literature, and a synthetic computational experiment.

11. REFERENCES

[1] D. Ward, "Enhancing Security: A Comprehensive Study on Deception Technology Integration in Manufacturing and Critical Infrastructure," Ph.D. dissertation, University of the Cumberlands, Williamsburg, KY, USA, 2025. Available: https://www.proquest.com/openview/ebf38e1aa599115548a9f7486917e669/1

[2] K. A. Stouffer, M. Pease, C. Y. Tang, T. Zimmerman, V. Y. Pillitteri, S. Lightman, A. Hahn, S. Saravia, A. Sherule, and M. Thompson, "Guide to Operational Technology (OT) Security," NIST Special Publication 800-82, Rev. 3, 2023. doi: 10.6028/NIST.SP.800-82r3.

[3] National Institute of Standards and Technology, "The NIST Cybersecurity Framework (CSF) 2.0," NIST Cybersecurity White Paper 29, 2024. doi: 10.6028/NIST.CSWP.29.

[4] International Electrotechnical Commission, "IEC 62443-2-1:2024, Security for Industrial Automation and Control Systems - Part 2-1: Security Program Requirements for IACS Asset Owners," 2024. Available: https://webstore.iec.ch/en/publication/62883

[5] Cybersecurity and Infrastructure Security Agency, "Cross-Sector Cybersecurity Performance Goals, Version 2.0," 2025. Available: https://www.cisa.gov/cross-sector-cybersecurity-performance-goals

[6] MITRE, "ATT&CK for ICS Matrix," ATT&CK, 2026. Available: https://attack.mitre.org/matrices/ics/

[7] J. Franco, A. Aris, B. Canberk, and A. S. Uluagac, "A Survey of Honeypots and Honeynets for Internet of Things, Industrial Internet of Things, and Cyber-Physical Systems," IEEE Communications Surveys & Tutorials, vol. 23, no. 4, pp. 2351-2383, 2021. doi: 10.1109/COMST.2021.3106669.

[8] S. Maesschalck, V. Giotsas, B. Green, and N. Race, "Don't

Get Stung, Cover Your ICS in Honey: How Do Honeypots Fit within Industrial Control System Security," Computers & Security, vol. 114, art. 102598, 2022. doi: 10.1016/j.cose.2021.102598.

[9] E. Lopez-Morales, C. Rubio-Medrano, A. Doupe, Y. Shoshitaishvili, R. Wang, T. Bao, and G. J. Ahn, "HoneyPLC: A Next-Generation Honeypot for Industrial Control Systems," in Proc. ACM SIGSAC Conf. Computer and Communications Security, 2020, pp. 279-291. doi: 10.1145/3372297.3423356.

[10] M. Lucchese, F. Lupia, M. Merro, F. Paci, N. Zannone, and A. Furfaro, "HoneyICS: A High-Interaction Physics-Aware Honeynet for Industrial Control Systems," in Proc. 18th Int. Conf. Availability, Reliability and Security, 2023, art. 113. doi: 10.1145/3600160.3604984.

[11] L. Salazar, E. Lopez-Morales, J. Lozano, C. Rubio-Medrano, and A. A. Cardenas, "ICSNet: A Hybrid-Interaction Honeynet for Industrial Control Systems," in Proc. Sixth Workshop on CPS&IoT Security and Privacy, 2024, pp. 68-79. doi: 10.1145/3690134.3694813.

[12] A. Javadpour, F. Ja'fari, T. Taleb, M. Shojafar, and C. Benzaid, "A Comprehensive Survey on Cyber Deception Techniques to Improve Honeypot Performance," Computers & Security, vol. 140, art. 103792, 2024. doi: 10.1016/j.cose.2024.103792.

[13] D. Ward, "Operationalizing Deception Technology in ICS/OT: A Control Mapping Framework for Critical Infrastructure Cybersecurity," International Journal of Soft Computing and Engineering, vol. 16, no. 3, pp. 1-9, 2026. doi: 10.35940/ijsce.C3723.16030726.

[14] D. Ward, "Decoy-Assisted Detection Metrics for Manufacturing Operational Technology Networks," International Journal of Scientific and Research Publications, vol. 16, no. 6, pp. 330-333, 2026. doi: 10.29322/IJSRP.16.06.2026.p17419.

[15] C. R. Harris et al., "Array Programming with NumPy," Nature, vol. 585, pp. 357-362, 2020. doi: 10.1038/s41586-020-2649-2.

[16] A. A. Hagberg, D. A. Schult, and P. J. Swart, "Exploring Network Structure, Dynamics, and Function Using NetworkX," in Proc. 7th Python in Science Conf., 2008, pp. 11-15. Available: https://networkx.org/