ISSN: 2598-8565 (media cetak)
ISSN: 2620-4339 (media online)

CHEATING DETECTION IN CAPTURE THE FLAG COMPETITIONS USING TWO-STAGE SIMILARITY ANALYSIS AND TIERED WEIGHTED RISK SCORING

Dimas Maulana✉, I Wayan Candra Winetra, I Nyoman Rai Widartha Kesuma
Undergraduate Applied Program in Software Engineering Technology, Department of Information Technology, Politeknik Negeri Bali, Badung, Indonesia
Email: dimasmaulana0305@gmail.com

DOI: https://doi.org/10.46880/jmika.Vol10No1.pp366-372

ABSTRACT

Flag sharing, inter-team cooperation (teaming), and prohibited tools such as AI undermine how validly a Capture the Flag (CTF) event measures skills, corrupting the integrity of its scoring information system. GZCTF's only cheating signal is the dynamic flag that catches flag theft. Nothing else feeds a per-team risk profile. This study aims to add a detection module to the GZCTF backend built on two components. The first is Two-Stage Similarity Analysis: pairwise Longest Common Subsequence and Jaccard scores are blended into a Relative Sequence Index (RSI), after which Confidence Screening Detector (CSD) screening confirms suspicious groups. The second is a tiered Weighted Risk Scoring model that assigns 38 indicators to four evidence tiers (Hard, Strong, Behavioral, Context), caps every non-Hard tier, and gives network or identity correlations no direct score. Evaluation used a controlled simulation of ten participations on a live instance, with design-time labels. Precision reached 1.000 with zero false positives, accuracy 0.900 and F1 0.909, and recall 0.833. An RSI threshold of 0.85 split every colluding pair from benign ones, and teams with purely network or identity correlation scored zero. The contribution is a tiered, capped aggregation model that stops weak signals from becoming false positives.

Keyword: Cheating Detection, Information System Integrity Audit, Capture the Flag, Weighted Risk Scoring, Collusion.

INTRODUCTION

Capture the Flag (CTF) competitions are a primary instrument for assessing cybersecurity skills (Balon & Baggili, 2023; Švábenský et al., 2021). The validity of that assessment is threatened by three cheating patterns: inter-team flag sharing, collusion on challenge solving, and the use of automated tools, including artificial-intelligence agents whose impact on CTF competitions has been documented (Pieterse, 2024). The open-source organizing platform GZCTF (GZTimeWalker, 2024) provides submission logs and dynamic flag detection, but it does not connect indicators across the network, behavioral, and similarity domains into a per-team risk profile, so investigation relies on manual log tracing that is impractical for competitions with hundreds of participants. For automation- and AI-assisted cheating in particular, current community practice still leans on manual measures. Organizers may ask a solving team to explain their steps while sharing their screen, or scrutinize write-ups for AI-generated traces, the latter effective only for highly technical challenges. Both scale poorly to hundreds of participants.

Collusion-detection research on online assessment has produced sequence- and set-similarity methods: the Longest Common Subsequence (LCS) for order similarity (Greenberg, 2002), the Jaccard index for set similarity (Travieso et al., 2024), and the Confidence Screening Detector (CSD) for group-level collusion screening (Xu et al., 2023). Data-mining approaches (Langebein et al., 2023) and clustering (Peng, 2024) reinforce the same direction, while cheating detection specific to CTF has begun to be explored through automated threat hunting (Chetwyn & Erdődi, 2021) and prevention via dynamic problem generation (Burket et al., 2015). On the identity side, browser fingerprinting provides a secondary signal (Laperdrix et al., 2020; Lin et al., 2023), and honeypots give a strong signal against automated scanning (Provos & Holz, 2007; Spitzner, 2002). To the author's knowledge, no prior work integrates these methods into a production CTF platform with an aggregation scheme that does not penalize honest teams. The closest approach, CTF Querier (Chetwyn & Erdődi, 2021), works post-competition over HTTP logs by tracing solving prerequisites backward from the flag, so it neither scores cross-domain risk in read-time nor filters network false positives as this system does.

Two gaps remain unaddressed by prior work. The first is integration, fusing network, behavioral, and

similarity evidence into one per-team risk profile. The second is aggregation. Under naive weight summation, an accumulation of low-confidence signals, for example shared IP addresses on a campus network, can exceed the score of a flag thief. This study adapts the risk-weighting practice of security engineering, namely CVSS vulnerability severity scoring (FIRST, 2019) and the NIST SP 800-30 risk assessment framework (National Institute of Standards and Technology, 2012), into a tiered Weighted Risk Scoring (WRS) model with a contribution cap per evidence tier.

This study makes three practical contributions and one theoretical contribution. First, it integrates Two-Stage Similarity Analysis, that is pairwise similarity validated by group screening, into the production GZCTF backend. Second, it presents a tiered WRS model whose four evidence tiers guarantee that a team without hard evidence cannot occupy the top rank. Third, it offers a quantitative evaluation through a controlled simulation on a live deployment instance, including a threshold sensitivity analysis and a single false-negative case that exposes the limit of evidence attribution in pairwise collusion. The theoretical contribution is to frame cheating detection as a read-time integrity audit over the scoring information system, in which a tiered, band-first model turns scattered logs into auditable per-team risk evidence and keeps low-confidence signals from accumulating into false positives.

RESEARCH METHOD

Two-Path Architecture on GZCTF

The detection module is embedded in the service and controller layers of GZCTF (C#/.NET 8, PostgreSQL) as two paths. The synchronous path captures submission metadata (the IP address, user agent, and timestamp) without delaying flag validation. The cheating analysis runs asynchronously on a queue worker after the data is stored, so detection adds no latency to the participant request. Evidence is stored in dedicated entities (SuspicionEvent, CheatInfo, ContainerAccessEvent, and AntiCheatBlock as a preventive login-block log) kept separate from the core transaction tables. The implementation is openly available as an open-source fork of GZCTF (Maulana, 2026).

Two-Stage Similarity Analysis

The first stage computes pairwise similarity. For two teams A and B, order similarity is taken from LCS (Greenberg, 2002) over the sequence of solved challenge identifiers and normalized by the shorter sequence length. Set similarity is computed with the Jaccard index over the set of solved challenges (Travieso et al., 2024). The two are combined into a Relative Sequence Index: $RSI(A,B) = 0.7 \cdot J(A,B) + 0.3 \cdot SLCS(A,B)$. The 70:30 weighting prioritizes set similarity, which is more resistant to submission-order manipulation. Combining an order-sensitive measure (LCS) with an order-insensitive one (Jaccard) captures both the sequence and the set of solved challenges, which neither measure alone can. To keep this order signal reliable, the participant interface presents challenges in a per-team randomized order (a deterministic Fisher-Yates shuffle seeded by the team rank id, shuffled per category), so that order similarity between two teams cannot be explained by both following the same on-screen layout. The second stage works at the group level with the CSD approach (Xu et al., 2023): starting from the highest-RSI pair, candidate-group membership is expanded step by step, and a new candidate is admitted only if its mean RSI against all existing members still exceeds the 0.75 cut-off. Incremental screening is used instead of group-level clustering (Peng, 2024) because it starts from the highest-confidence pair and leaves an auditable membership trail.

Tiered Weighted Risk Scoring

Thirty-eight indicators are mapped to four evidence tiers: Hard, for example flag theft, dynamic flag leakage, a canary flag on a honeypot, and cross-team container access, with no cap; Strong, for example solution relay and machine-speed submission patterns, with a subtotal cap of 60; Behavioral, comprising timing heuristics, with a cap of 25; and Context, comprising network and identity correlations whose direct score is always zero. A team's total score is $Stotal = SHard + Scorr + SStrong + SBehavioral$, where $SStrong = \min(60, \Sigma \text{ weights})$ and $SBehavioral = \min(25, \Sigma \text{ weights})$ are computed after the per-rule incident cap is applied, and the corroboration $Scorr = \min(SHard/2, \Sigma \text{ Context units})$ is active only when Hard evidence is present. The non-Hard subtotal is capped at 85. Ranking is decided by band first, namely Evidenced (has Hard evidence), Investigate (Strong), Watch (Behavioral), Context, then Clean, and only then by score within the band. As a result, no stack of network signals, however large, can make an honest team appear as a principal offender. The score is recomputed each time it is read from the evidence history rather than accumulated, so changes to rule weights take effect immediately without data migration.

Table 1 lists all 38 indicators under their evidence tier, with each tier's subtotal ceiling, risk

band, and the default weight of every indicator, so a per-team total such as 270 (Team 07) can be traced to its evidence. Because a Hard signal places a team in the Evidenced band regardless of magnitude, no accumulation of network-tier signals can outrank an actual flag thief. The detection logic and thresholds behind each rule are in the open-source repository (Maulana, 2026).

Table 1. All 38 Indicators Across the Four Evidence Tiers (Subtotal Ceiling, Risk Band, Default Weight)

Tier	Ceiling	Band	Indicators (default weight)
Hard	none	Evidenced	CrossTeamContainerAccess (120), StolenFlag (100), HoneypotCanaryFlag (100), TokenAbuse (80), WrongFlagLeakage (80)
Strong	60	Investigate	HoneypotChain (150), HoneypotProtocolHit (90), SolutionRelay (60), AutomatedPattern (50), HighWrongRate (40)
Behavioral	25	Watch	NoDownload (80), NoContainer (80), HoneypotHit (70), AdaptiveFastSolve (60), FastSolve-Open (50), FastSolve-Download (50), FastSolve-Container (50), ZeroWrongAttempts (50), InstantSubmitAfterAccess (50), SequenceSimilarity (40), DelayedSolveSubmission (40), Burst (30), Hoarding (30), DirectedSolving (30), SubmitterNeverAccessedContainer (30), FirstBloodAnomaly (20), CollusionGroup (10)
Context	0 (corroborates Hard)	Context	FlagEgress (80), SharedFingerprint (60), ClusteredRegistration (40), FingerprintChurn (30), SessionConcurrency (30), AccessIpMismatchAtSubmission (30), CrossTeamIP (20), IpChurn (20), SharedIP (10), UnknownIP (10), SubnetOverlap (5)

Evaluation Design

Because real competition logs labelled by an organizing committee were unavailable, accuracy was measured through a controlled simulation on a public deployment instance. The test game contained eleven participations, but the platform admin account, a non-competitor control, was excluded from the evaluation population, leaving ten participations whose labels were fixed at design time by construction. A participation counts as positive (cheating) when a cheating scenario was injected into it and negative (benign) otherwise. Six participations ran five kinds of cheating scenario (flag theft; an automated scanner that triggers the honeypot; a leader-follower collusion pair; dynamic flag leakage; and container-access anomalies), and four others behaved legitimately or carried noise that commonly triggers false alarms (a legitimate fast team; identity churn; and shared IP and subnet as on campus NAT). Classification verdicts were computed by the actual scoring engine. A participation in band Investigate or above was classified as detected. Metrics were computed from the confusion matrix: accuracy, precision, recall, F1, specificity, and FPR (Sokolova & Lapalme, 2009). The sensitivity of the RSI threshold, the combining weights, and the CSD cut-off was tested over all 45 participation pairs. The design assumes a single live instance and one game, scenarios authored to represent each cheating family rather than sampled from the field, and the default rule weights of Table 1 held fixed. Report latency was measured two ways: twenty sequential requests on the public deployment instance and an integration-suite benchmark on synthetic competitions of 12, 24, and 48 teams. As an additional data source, organic traffic on the same instance before the scenarios were injected was observed to test the fairness property under real conditions.

RESULTS AND DISCUSSION

Detection Results on the Controlled Simulation

Table 2 shows the ten simulated participations with their labels and the system verdict on each. The five participations carrying Hard or Strong evidence occupy the Evidenced band (scores 105 to 270) and Investigate (85), while none of the four benign participations rose past the Watch band. The final result is TP = 5, FP = 0, FN = 1, and TN = 4 (Figure 1).

Table 2. Ten Simulated Participations: Injected Scenario, Label, and System Verdict

Participation	Injected scenario	Label	Band (score)	Result
Team 07	Steals another team's flag, shares token, machine-fast submit rhythm	Cheating	Evidenced (270)	TP
Team 06	Automated scan: traverses the honeypot bait chain to the canary flag	Cheating	Evidenced (195)	TP
Team 03	Accesses another team's container	Cheating	Evidenced (150)	TP

ISSN: 2598-8565 (media cetak)
ISSN: 2620-4339 (media online)

Participation	Injected scenario	Label	Band (score)	Result
Team 08	Submits another team's dynamic flag as the answer; repeated guessing	Cheating	Evidenced (105)	TP
Team 10	Collusion follower: mimics the solving order, same fingerprint	Cheating	Investigate (85)	TP
Team 04	Collusion leader (paired with Team 10)	Cheating	Watch (25)	FN
Team 09	Legitimate skilled team; speed anomaly only	Benign	Watch (25)	TN
Team 02	Identity churn plus mild behavioral noise	Benign	Watch (25)	TN
Team 01	Clustered registration and shared subnet, with fast-solve noise	Benign	Watch (25)	TN
Team 05	Plays normally	Benign	Watch (25)	TN

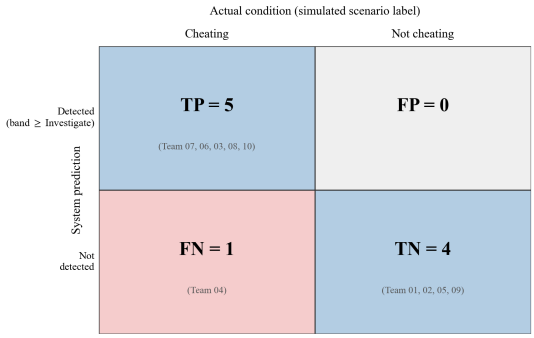

Figure 1. Confusion Matrix of the Controlled Simulation

The metric summary is given in Table 3: accuracy 0.900; precision 1.000; recall 0.833; F1 0.909; specificity 1.000; FPR 0.000. Full precision means all four benign participations passed without a suspect label, even though two of them were deliberately flooded with Context-tier signals. Community-relative suppression gates (excluding easy challenges, requiring a minimum number of solvers, and neutralizing cohorts) also hold back speed signals on benign teams.

Table 3. Summary of Detection Metrics

Metric	Value	Metric	Value
Accuracy	0.900	F1	0.909
Precision	1.000	Specificity	1.000
Recall	0.833	FPR	0.000

The case that slipped through is the collusion-pair leader: the solution-relay events and fingerprint match were recorded under the follower's name, leaving the leader with only Behavioral-tier evidence, which was capped at 25 points and stopped at the Watch band. Such lopsided evidence attribution is an inherent limit of pairwise collusion detection. An administrator can still trace it through the cross-evidence reference and the collusion-group panel. Copying evidence symmetrically to every group member is an improvement whose impact on recall can be measured.

Threshold Sensitivity

The RSI distribution over the 45 pairs shows a wide gap between colluding and benign pairs: the three highest values (1.000; 0.883; 0.883) form the Team 04, Team 10, and Team 07 cluster. Team 07 is the third node because its solving order was deliberately set to be similar. The highest benign pair reaches only 0.767. Table 4 tests three candidate thresholds: at 0.90 only the core pair remains so the cluster breaks; at 0.75 one benign pair begins to be caught; at 0.85 all three colluding pairs are caught without bringing in any benign pair. The group mean RSI (0.922) is also well above the CSD cut-off of 0.75, so the 0.85/0.75 threshold pair used by the system is confirmed. A sweep of the combining weights (70:30, 60:40, 50:50) over the same formula and group-growth algorithm shows that the 0.85 threshold still separates perfectly in all three. The widest separation margin actually appears at 70:30 (0.117 versus 0.100 and 0.083), because order-heavy weighting raises the score of benign pairs with short sequences. None of the nine combinations of weight and CSD cut-off (0.70/0.75/0.80) ever changed the composition of the cheating cluster.

Table 4. RSI Threshold Sensitivity Across 45 Pairs

Threshold	Pairs ≥ threshold	Collusion caught	Benign included
0.90	1	1 of 3	0
0.85	3	3 of 3	0
0.75	4	3 of 3	1 (0.767)

Fairness on Real Traffic

Before the scenarios were injected, the same instance ran organically and recorded nine events across four participations, all of them Context-tier indicators in the form of shared IP, subnet overlap, and new IP. The effective score of those four participations was zero, in band Context. Had the weights been summed naively, the four would have carried a total of 70 points and appeared, wrongly, as suspect teams. The same pattern is visible in the simulation: every benign participation stops exactly at the Behavioral cap of 25

ISSN: 2598-8565 (media cetak)
ISSN: 2620-4339 (media online)

points (band Watch), because behavioral noise is clipped by the cap rather than accumulated (Figure 2).

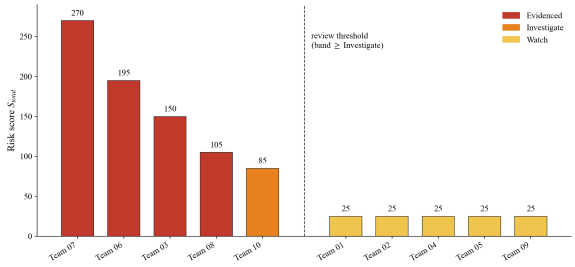

Figure 2. Risk-Score Distribution of All Participations, Grouped by Band

Deployment Evidence

Both components run on the live deployment instance, not only on test fixtures. Figure 3 shows the administrator monitoring dashboard, where each participation is ranked by risk band (Evidenced, Investigate, Watch) alongside summary counters for hard evidence, IP anomalies, abnormal solves, collusion, and identity overlap. Five of the six injected cheating participations occupy the Evidenced and Investigate bands, while the sixth (Team 04, the collusion leader whose evidence attributes to its follower) and all four benign participations stop at Watch, consistent with Table 2.

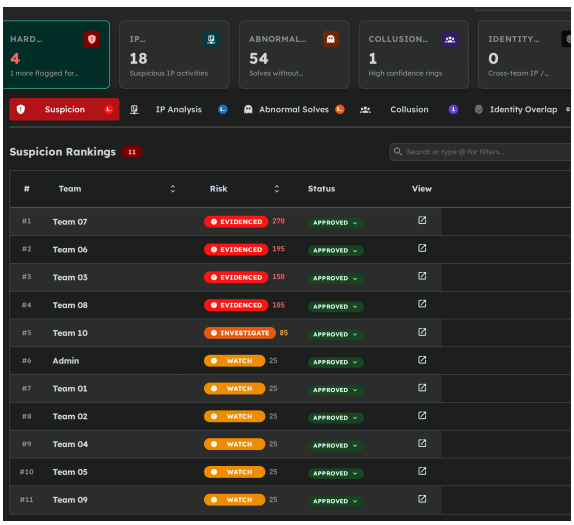

Figure 3. Administrator Anti-Cheat Dashboard: Per-Team Risk Bands and Evidence Summary (Row Admin is the Non-Competitor Control, Excluded from the Metrics)

Performance Within a Measured Scope

Because the analysis runs asynchronously on a queue worker (see the method section), participants incur no extra latency and the computation cost appears only when a report is read. Measurement on the public deployment instance shows that twenty sequential report requests yield a median of 212 ms and a p95 of 426 ms, network latency included, so the dashboard stays interactive for administrators. A controlled benchmark on the integration suite (a dedicated test PostgreSQL; each team carries 6 to 8 solves and a unique login log) gives a growth curve: warm median 89 ms at 12 teams, 201 ms at 24 teams, and 523 ms (p95 606 ms) at 48 teams, rising 2.3 to 2.6 times per doubling of the team count, consistent with the quadratic pairwise-comparison stage. A full load test requires an isolated staging environment and remains future work.

Position Relative to Related Work

The tiered scoring model in this system treats order similarity as one signal among 38 indicators, unlike test-collusion detection that makes response similarity the single signal. The group-growth algorithm inherits the idea of the Confidence Screening Detector (Xu et al., 2023), namely starting from a suspect pair and adding members as long as the mean similarity stays high. Peng (2024) examines clustering as an alternative for group-level collusion. Unlike the online-exam setting that relies on response patterns (Langebein et al., 2023), the CTF domain offers richer evidence in the form of network traces, container access, and downloads, so order similarity does not stand alone but is corroborated by other signals before raising the risk band. Against prior CTF cheating detection (Chetwyn & Erdődi, 2021) that traces HTTP logs post-competition, this system adds read-time cross-domain aggregation and a network-fairness gate that suppresses false positives from legitimate collaboration, as confirmed on the real-traffic test. The effect is concrete. Treating similarity as one tiered signal among many kept precision at 1.000 where single-signal response-similarity methods stay exposed to coincidental matches, and the fairness gate scored the four real shared-network participations at zero where a naive sum would have charged 70 points. This matches what those studies report. Peng (2024) finds that even the best-balanced clustering method reaches the lowest Type I (false-positive) error only by giving up detection power, and Langebein et al. (2023) calibrated against a proctored control group to decide which similarities count as suspicious. The tiered model here instead held the false-positive rate at zero on the controlled simulation.

Limitations

The evaluation dataset is a controlled simulation of ten author-designed participations, not a full committee-labelled competition; recall of 0.833 has not

reached the 0.90 target; and scoring still applies per competition, so the track record of repeat offenders across competitions is not yet linked automatically. Because there are only six positive and four negative cases, a single misclassification shifts a metric by about 16.7 points on recall or 25 points on FPR, so these figures are indicative for a controlled simulation rather than a population estimate. These limitations echo prior work. Labelled real-world CTF cheating data is scarce, so earlier CTF detection likewise relied on post-competition logs rather than a labelled benchmark (Chetwyn & Erdődi, 2021), and adjacent collusion studies evaluate on exam or test-response data (Langebein et al., 2023; Peng, 2024). The single false negative, a collusion leader whose evidence attached to its follower, reflects the asymmetric attribution that group-symmetric methods avoid (Xu et al., 2023).

CONCLUSION

A cheating-detection module based on Two-Stage Similarity Analysis and tiered Weighted Risk Scoring was successfully integrated into GZCTF without burdening the answer-submission path. In a controlled-simulation feasibility study, the system reached a precision of 1.000 and zero FPR with a recall of 0.833; an RSI threshold of 0.85 separated every colluding pair from benign ones; and teams that merely shared network infrastructure scored zero, a fairness property also confirmed on real traffic. Future work includes symmetric evidence propagation across a collusion group, linking evidence across competitions, validation on a labelled real competition, and a load test on a staging environment.

REFERENCES

Balon, T., & Baggili, I. (2023). Cybercompetitions: A survey of competitions, tools, and systems to support cybersecurity education. *Education and Information Technologies, 28*(9), 11759–11791. https://doi.org/10.1007/s10639-022-11451-4

Burket, J., Chapman, P., Becker, T., Ganas, C., & Brumley, D. (2015). Automatic problem generation for capture-the-flag competitions. *2015 USENIX Summit on Gaming, Games, and Gamification in Security Education (3GSE 15)*. https://www.usenix.org/conference/3gse15/summit-program/presentation/burket

Chetwyn, R. A., & Erdődi, L. (2021). Cheat detection in cyber security captures the flag games – An automated cyber threat hunting approach. *Proceedings of the 28th C&ESAR*, 175–190. https://ceur-ws.org/Vol-3056/paper-11.pdf

FIRST. (2019). *Common Vulnerability Scoring System version 3.1: Specification document*. https://www.first.org/cvss/v3.1/specification-document

Greenberg, R. I. (2002). *Fast and simple computation of all longest common subsequences* (arXiv:cs/0211001). arXiv. https://arxiv.org/abs/cs/0211001

GZTimeWalker. (2024). GZCTF: The GZ::CTF project, an open source CTF platform [Computer software]. *GitHub*. https://github.com/GZTimeWalker/GZCTF

Langebein, J., Massing, T., Klenke, J., Striewe, M., Goedicke, M., Hanck, C., & Reckmann, N. (2023). A data mining approach for detecting collusion in unproctored online exams. *Proceedings of the 16th International Conference on Educational Data Mining*, 6–16. https://doi.org/10.5281/zenodo.8115649

Laperdrix, P., Bielova, N., Baudry, B., & Avoine, G. (2020). Browser fingerprinting: A survey. *ACM Transactions on the Web, 14*(2), 1–33. https://doi.org/10.1145/3386040

Lin, X., Araujo, F., Taylor, T., Jang, J., & Polakis, J. (2023). Fashion faux pas: Implicit stylistic fingerprints for bypassing browsers' anti-fingerprinting defenses. *2023 IEEE Symposium on Security and Privacy (SP)*, 987–1004. https://doi.org/10.1109/SP46215.2023.10179437

Maulana, D. (2026). GZCTF with integrated cheating detection [Computer software]. *GitHub*. https://github.com/dimasma0305/GZCTF

National Institute of Standards and Technology. (2012). *Guide for conducting risk assessments (NIST Special Publication 800-30 Rev. 1)*. https://doi.org/10.6028/NIST.SP.800-30r1

Peng, L. (2024). Comparing clustering methods in group-level test collusion detection. *Proceedings of the 17th International Conference on Educational Data Mining,* 893–897. https://doi.org/10.5281/zenodo.12729989

Pieterse, H. (2024). Friend or foe – The impact of ChatGPT on capture the flag competitions. *International Conference on Cyber Warfare and Security, 19*(1), 268–276. https://doi.org/10.34190/iccws.19.1.1992

Provos, N., & Holz, T. (2007). *Virtual honeypots: From botnet tracking to intrusion detection*. Addison-Wesley.

Sokolova, M., & Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Information Processing & Management, 45*(4), 427–437. https://doi.org/10.1016/j.ipm.2009.03.002

Spitzner, L. (2002). *Honeypots: Tracking hackers*. Addison-Wesley.

Švábenský, V., Čeleda, P., Vykopal, J., & Brišáková, S. (2021). Cybersecurity knowledge and skills taught in capture the flag challenges. *Computers & Security, 102*, 102154. https://doi.org/10.1016/j.cose.2020.102154

Travieso, G., Benatti, A., & Costa, L. da F. (2024*). An analytical approach to the Jaccard similarity*

ISSN: 2598-8565 (media cetak)
ISSN: 2620-4339 (media online)

index (arXiv:2410.16436). arXiv. https://arxiv.org/abs/2410.16436

Xu, Y., Cui, Y., Wang, X., Huang, M., & Luo, F. (2023). Confidence screening detector: A new method for detecting test collusion. *Applied Psychological Measurement, 47*(3), 237–252. https://doi.org/10.1177/01466216231165299