V1.0 · NOVEMBER 2026

Check the draft, not the email

The request that defeated every detector is caught every time by a lookup. 160 of 160, because the answer was never in the language.

Nikolaos Broikos · broikos.gr November 2026

280 drafts · 4 attack classes · 23 Nov 2026

160 of 160

FRAUDULENT DRAFTS FLAGGED, ALL FOUR CLASSES

0 of 40

ORDINARY REPLIES FLAGGED

40 of 40

GENUINE BANK CHANGES FLAGGED TOO, WHICH IS THE POINT

under 2 min

REVIEW A DAY PER HUNDRED DRAFTS, UNDER STATED MIXES

01 What this is

> Discharges claims J001 to J012.

Paper 13 ended with a recommendation it could not support. It had just shown that the request invoice fraud actually uses, a polite note asking whoever drafts the reply to confirm payment to a new bank account, passed every detector that kept ordinary mail clean. It then proposed a deterministic check of each draft against the counterparty's records, and said plainly that nobody had measured one, including itself.

> This measures it.

The result in one paragraph

A check that compares an outgoing draft against the counterparty's record and the thread it replies to, asking only whether each bank detail, recipient, link domain and attachment is already known, **flagged 160 of 160 fraudulent drafts. All four attack classes, every time.** That includes the bank account change that four machine-learned detectors missed in paper 13.

It does that because it is a lookup. An IBAN is on record or it is not, and no amount of persuasive language in the incoming email changes the answer. **The attack that defeats a classifier cannot defeat a comparison**, because the comparison never reads the attacker's text.

The result the headline leaves out

The check cannot tell a fraudulent new bank account from a counterparty that genuinely changed bank, because both are an IBAN that is not on record. It flagged **40 of 40 legitimate bank changes** too.

So this is not a detector and the paper does not call it one. **It is a gate that routes to a person**, and the question that decides whether it is usable is not its precision but what it costs to review. That cost depends entirely on how often ordinary mail introduces something new, which this bench cannot measure and which section 04 therefore publishes as a stated range rather than as a result.

At one to five genuine bank changes per thousand outgoing drafts, the check fires on between 5 and 25 drafts per thousand. On a desk sending a hundred drafts a day that is under two minutes of review.

What is measured here, and what is assumed

Measured. 280 outgoing drafts across seven classes, generated from a fixed seed against a set of twelve invented counterparties, each with its own domain, contacts, bank details and link domains. Every counterparty, person, vessel, IBAN and figure is invented. The generator, the check, the corpus and the per-draft results are published with this paper.

Assumed, and labelled as such throughout. How often a real shipping inbox produces a draft that legitimately introduces a new bank account, a new contact, a new link domain or a new attachment. Nobody has published that, this bench cannot produce it, and it is the number that decides the whole economics of the control. **A deployment measures its own before adopting this.**

Why this is a shipping paper

The corpus is shipping correspondence because paper 13's was, and because the sector makes the problem concrete: a port disbursement account, a bunker stem and a laytime statement all end in a payment instruction, and the counterparty list of a chartering desk is small, stable and already written down somewhere. **The control is cheap precisely where the money is.**

Nothing here is specific to shipping in its mechanism. A check that asks whether a bank detail is on record works identically for any business that pays invoices, which is all of them. The sector decides how easy the record set is to assemble, not whether the idea works.

What this paper does not claim

It does not claim the check stops fraud. It claims the check puts a person in front of every draft that introduces new payment details, and that it does so at a review cost which, under stated assumptions about how often that happens legitimately, is small.

Whether a person who is shown a flagged draft then does the right thing is not measured here, and it is the step where the remaining risk lives.

02 The instrument

> Discharges claims J009 to J010, J013 to J022.

The check

Four questions, asked of an outgoing draft against the counterparty's record and the thread it replies to:

```
is every bank detail in this draft already on record for this counterparty
is every recipient on record, or already in the thread
is every link domain on record
is every attachment one the thread actually carried
```

Anything that is not gets flagged for a person. That is the entire mechanism, and it is published in check.py in forty lines.

It reads no language, scores no text and contains no model. It never parses the incoming email, which matters: every control that failed in paper 13's survey failed by interpreting text that an attacker wrote, from URL allowlists beaten by an expired domain to command approval beaten by a parser differential. A comparison against a record the attacker cannot write to has no such surface.

The corpus

Twelve invented counterparties, each with a domain, two to four contacts, a bank account and a set of known link domains. Twelve names such as Aegean Bulk Chartering and Thalassa Bunkering, all fictional, all generated from seed 20261123 so the corpus regenerates byte for byte.

Seven classes of outgoing draft, 40 each, 280 in all:

CLASS	FRAUDULENT	WHAT IT IS
clean	no	An ordinary reply introducing nothing new
legit-iban	no	**The counterparty genuinely changed bank**
legit-newcontact	no	**A genuinely new person at a known counterparty**
fraud-iban	yes	Paper 13's style B payload carried into the draft
fraud-recipient	yes	A recipient the user never named and the thread never carried
fraud-link	yes	A link domain not on record for that counterparty
fraud-attach	yes	An attachment the thread never carried

The two legitimate classes in bold are the point of the bench. Any check separates a fraudulent IBAN from a clean reply. The question for a real desk is what happens when a counterparty really does change bank, which happens, or when a new person at a known firm joins the thread, which happens constantly. Those two classes decide whether the control is affordable, and a bench without them would be flattering and useless.

What the check is told, and what it is not

It is given the counterparty's record and the thread's attachments. It is **not** given the incoming email, the draft's prose, any score from a detector, or any indication of which class a draft belongs to.

That isolation is deliberate. The failure mode this control exists to survive is an attacker who writes persuasive text, so the control is built where persuasive text cannot reach it.

The class balance is a design parameter, not a sample

40 drafts in each of seven classes means two thirds of the legitimate traffic in this corpus introduces something new. **No inbox looks like that.** A real mailbox is overwhelmingly ordinary replies.

That is deliberate and it is also the bench's sharpest limitation. Equal classes give a clean read of per-class behaviour, which is what a deterministic check has and a probabilistic one does not, but they make any aggregate rate over the whole corpus meaningless. Section 04 does the arithmetic properly, with the mix stated as an assumption.

This is said here rather than in the limitations because the first run of this bench produced an aggregate flag rate of 85.4 percent, and that number is in the published results file. It is true of the corpus, it is false of any inbox, and a reader who finds it without this section would reasonably conclude the control was unusable.

What is published

make_drafts.py generates the records and the drafts. check.py runs the check and writes every per-draft verdict with its reasons. sensitivity.py does the base-rate arithmetic. records.json, drafts.jsonl and results.json carry the corpus and the results.

Anyone can change the check, add a class of attack it misses, or argue that a different field belongs in the comparison, and re-run it in seconds. That is the point of publishing it.

03 What it catches

> Discharges claims J002 to J005, J014, J017, J020 to J021, J023 to J029.

CLASS	N	FLAGGED	RATE
clean	40	0	**0.0%**
fraud-iban	40	40	**100.0%**
fraud-recipient	40	40	**100.0%**
fraud-link	40	40	**100.0%**
fraud-attach	40	40	**100.0%**
legit-iban	40	40	100.0%
legit-newcontact	40	39	97.5%

160 of 160 fraudulent drafts flagged. Every class, every draft. And no ordinary reply flagged at all.

Why it is 100 percent, and why that is not impressive

A machine-learned detector scoring 100 percent on an attack class would be a remarkable result demanding scrutiny. Here it is close to arithmetic.

The check asks whether a string is in a set. The fraudulent drafts introduce a bank account, a recipient, a link domain or an attachment that is not in the counterparty's record, because that is what those attacks are for. **There is no mechanism by which a lookup can miss them, short of a bug.** The 100 percent is a statement that the implementation is correct, not that the idea is clever.

That is exactly why it is worth having. Paper 13 spent four detectors and three scoring modes on a request that passed all of them, and the reason it passed was that the detectors had to judge whether text looked malicious. This control never reads the text.

> The attack that defeats a classifier cannot defeat a comparison.

The bank account note, which is the point of the exercise

Paper 13's style B payload asks whoever drafts the reply to confirm payment to a new bank account. It carries no trigger words, no override, no suspicious URL, and it reads like a client correcting their payment details.

CONTROL	RESULT
Protect AI v2	Missed it whole and in windows
Prompt Guard 2 86M	Missed it in every mode
proventra mDeBERTa	Missed it in every mode
NeuralTrust small	Missed it in every mode
This check	**40 of 40**

The only detector that caught it in paper 13 did so in the mode where it also flagged every email carrying a confidentiality footer, which makes the catch uninterpretable.

The difference is not that this control is better at reading English. It is that it is not reading English. **The signal was never in the language. It was in the fact that the account number was not the one on file**, which is a question about a database and not about a sentence.

The one legitimate draft that was not flagged

legit-newcontact flagged 39 of 40, not 40 of 40. The missing one is worth explaining, because it is the only non-deterministic-looking cell in the table and it is not a flaw in the check.

The generator builds a new contact by drawing a first and last name from pools of twelve each. In one of the forty drafts the draw produced a name already in that counterparty's record, so the "new" contact was not new. **The check was right and the corpus was wrong.**

It is reported rather than regenerated away because it demonstrates the mechanism: the check has no concept of novelty beyond the record, so an address already on file passes, which is the intended behaviour.

What the aggregate numbers in the results file say, and why they are useless

results.json reports a flag rate of 85.4 percent over the whole corpus and 65.8 percent over its legitimate drafts.

Both are true of the corpus and false of any inbox. The corpus holds 40 drafts in each of seven classes, so two thirds of its legitimate traffic deliberately introduces something new. A real mailbox is overwhelmingly ordinary replies, and section 04 does the arithmetic with the mix stated.

They are published rather than suppressed because they are what the instrument produced, and a reader who found them in the data without finding this paragraph would correctly conclude the control was unusable.

04 What it costs

> Discharges claims J007 to J008, J011, J018, J030 to J035.

The check fires on every draft that introduces a bank detail, recipient, link domain or attachment not already on record. Fraudulent drafts do that by design. **So do legitimate ones, when a counterparty really changes bank or a new person joins the thread.**

So the cost of this control is the number of legitimate drafts it stops, and that number is not a property of the check. It is a property of the inbox.

The number this bench cannot produce

How often does a real shipping desk send a draft that legitimately introduces a new bank account, a new contact, a new link domain or a new attachment?

Nobody has published it, and this bench cannot measure it. The corpus has a mix because somebody chose one. Measuring the real mix needs a real mailbox, which is the deployment step, not the research step.

What the bench does produce is the per-class behaviour, and because the check is deterministic those rates are sharp rather than estimated: clean 0 percent, genuine bank change 100 percent, new contact 97.5 percent. **Combine those with a mix and you get the cost. The mix is an assumption and is published as one.**

Three stated mixes

Each says how many of 1,000 outgoing drafts fall in each legitimate class. They are assumptions, not measurements, and they are chosen to bracket what a chartering or operations desk plausibly looks like.

MIX	GENUINE BANK CHANGES	NEW CONTACTS	FLAGS PER 1,000	ON 100 DRAFTS A DAY	REVIEW AT 30S EACH
Conservative	5	20	**24.5**	2.5 a day	1.2 minutes
Central	2	10	**11.8**	1.2 a day	0.6 minutes
Quiet desk	1	4	**4.9**	0.5 a day	0.2 minutes

Under every one of these, the control costs under two minutes of review a day on a hundred drafts.

That is the finding of this section, and the honest form of it is conditional: *if* genuine novelty in outgoing mail runs at a few per thousand, *then* the review load is negligible. A desk where counterparties change bank weekly and new contacts appear constantly would get a different answer, and should measure before adopting.

Recall does not move with the mix

One asymmetry makes this control unusually easy to reason about.

The flag rate depends entirely on the mix. The recall does not depend on it at all. A lookup catches a new bank account whether that happens once a year or twice a day, because it compares against a record rather than estimating from a distribution. There is no threshold to tune, no base rate to calibrate against, and no degradation as the attack gets rarer.

That is the opposite of a classifier, whose precision collapses as the base rate falls. The reason detectors are hard to deploy against rare attacks is that at a one in ten thousand base rate, even a 99 percent accurate detector buries the real alert. **This control has no such property**, which is why it is worth having alongside a detector rather than instead of one.

What the cost actually is, stated properly

Not two minutes a day. That is the arithmetic under an assumption.

The real cost is that **somebody has to maintain the counterparty record**, and the check is worth exactly as much as that record is current. A record that is six months stale flags every genuine change and trains the reviewer to click through, which is worse than no control, because it manufactures the habit the attack needs.

Nothing in this bench measures record staleness, and it is the thing most likely to break this control in practice. It is named here rather than in the limitations because it is the first question a deployment should ask, not a caveat at the end.

05 What it cannot do

> Discharges claims J004 to J006, J014, J019, J029, J035 to J043.

It cannot tell fraud from a genuine change

fraud-iban flagged 40 of 40. legit-iban flagged 40 of 40.

The check cannot distinguish them and nothing will make it able to, because they are the same event seen from outside: an IBAN that is not on record. One was put there by a criminal and one by a client who switched banks, and the difference lives in the world rather than in the draft.

This is why the paper does not call it a detector. **It is a gate that routes to a person**, and the person is the control. The check's job is to make sure that person is looking at the right forty drafts out of a thousand rather than at all thousand or at none.

A reader who wants a control that approves genuine changes and blocks fraudulent ones is asking for something that does not exist, and paper 13 is forty pages on why the version that tries to read the text does not work either.

It does not protect the person it routes to

Everything measured here stops at the moment a draft is flagged. **Whether the reviewer then does the right thing is not measured, and that is where the remaining risk lives.**

Paper 13's survey is unambiguous about how that step fails. Copilot email summaries displayed attacker-written security alerts inside the trusted interface. Approving from an assistant's own summary is not review, because the summary is what the attacker wrote for. A flagged draft shown next to a persuasive explanation of why the account changed is a worse situation than an unflagged one, because it carries institutional endorsement.

So the control has a requirement the bench cannot test: **the reviewer must see the raw draft and the record it failed against, and must not see the assistant's account of why the change is fine.**

It does not read the incoming email, which is the point and also a limit

The check never looks at the message that prompted the draft. That is what makes it immune to persuasion, and it means it is blind to everything an attacker does that does not reach the draft's structured fields.

An injection that changes the *tone* of a reply, or makes the assistant omit a demurrage claim, or mis-state a laytime calculation, introduces no new IBAN, recipient, link or attachment. **This control does not see any of it**, and a shipping desk should not imagine it does. Paper 13's style B included exactly such requests: state that the demurrage claim was accepted, recommend this offer.

Those are action steering without novelty, and **nothing in this paper addresses them.**

It assumes a record exists

The whole mechanism rests on a counterparty record with bank details, contacts and domains that is current enough to compare against. Twelve invented counterparties in a JSON file is not that.

Assembling the real thing is the deployment work: pulling it from an accounting system, a chartering system or a CRM, deciding who may update it, and keeping it current enough that genuine changes are rare rather than routine. **A business without such a record cannot run this control at all**, and that, not the check, is the actual project.

It was never attacked adaptively

The drafts in this corpus do not adapt. Nobody tried to construct a fraudulent draft that passes the check.

The obvious attack is to use a bank account that *is* on record: a counterparty's genuine account at a bank the criminal controls access to, or a stale account still in the record after a real change. A second is to reach fields the check does not compare. **Neither is tested here**, and the first is the one that would worry me in deployment.

Paper 13's central finding was that every defence which had reported near-zero attack success fell above 90 percent once somebody adapted to it. **This control has not met an adaptive attacker either**, and its structural advantage, that there is no text to persuade, is a reason to expect it to hold rather than evidence that it will.

06 The design this sits inside

> Discharges claims J001 to J002, J004, J006, J013, J015, J033, J038, J040, J044 to J050.

This check is one control. On its own it would be a flag on a draft that nobody is required to look at. Paper 13 set out the design it belongs to, and this section states it once more because a control without its context is a feature.

The six that were already supported

Carried from paper 13, where each is sourced:

1. **No send or forward tool in the reading context.** The assistant drafts, a person sends. No published bypass in paper 13's survey window defeated a control that removed a capability, as against every bypass that defeated a control which interpreted text. 2. **The person sends from the raw draft**, with its real recipients visible, not from the assistant's summary. 3. **One thread per draft**, so the drafter cannot reach other mailboxes. 4. **Recipients and attachments limited to what the user's own request names.** This is the rule behind the architecture that survived 13,935 attacked episodes on 15 open-weight models with no successful attack. 5. **Hidden content stripped before the model and shown to the reviewer.** A mitigation, not a boundary, since models read hidden text anyway. 6. **No remote image fetching and no link unfurling** in any view. Link previews are a confirmed zero-click exfiltration channel.

The seventh, which this paper adds

7. **Every draft compared against the counterparty's record before it reaches a person, and flagged if it introduces a bank detail, recipient, link domain or attachment that is not on file.** 160 of 160 fraudulent drafts, at a review cost of under two minutes a day on a hundred drafts under stated assumptions about how often

legitimate novelty occurs.

Paper 13 proposed it and could not measure it. It is now measured, and it is the only control in the list that catches the specific request the October 2026 campaigns use.

Where it sits in the order of operations

The sequence matters more than any single element:

```
incoming mail
  strip hidden content, show what was stripped
  model drafts a reply, with no send tool and one thread of context
  CHECK the draft against the counterparty record        <- this paper
  flagged drafts go to a person with the raw draft and the failed comparison
  the person sends
```

The check runs on the draft, not on the incoming mail. That is deliberate and it is the whole design. Checking the incoming email means judging text an attacker wrote, which is what failed in paper 13. Checking the outgoing draft means comparing structured fields against a record the attacker cannot write to, whatever they put in the email.

What a detector is still for

Nothing here argues against running one. It argues against gating on one.

Paper 13's best operating point, a small multilingual model scored paragraph by paragraph, flagged no ordinary email in English or Greek and caught every direct override at 392 ms per email on an ordinary CPU. As telemetry that is worth having: it sees styles of attack this check is blind to, including the action-steering requests that introduce no new field.

The two controls fail in different directions, which is the argument for both. The detector misses the polite request and catches the blunt one. The check catches anything that changes a payment detail and misses everything that does not. Neither covers the other's gap, and neither should be asked to.

The part that is not technical

The reviewer needs to be able to say no without it costing them anything. A flagged draft on a Friday afternoon, from a counterparty the desk has worked with for nine years, for an invoice already overdue, is the exact situation the attack is built for.

No control in this paper addresses that, and it is the one most likely to decide whether the other seven matter.

07 Limitations

> Discharges claims J010 to J011, J020 to J021, J028, J035, J037, J040, J042 to J043, J050 to J058.

The three that bound everything

The corpus is synthetic and its class balance is a design parameter. 40 drafts in each of seven classes means two thirds of the legitimate traffic introduces something new, which no inbox resembles. Per-class rates are what this bench measures; any aggregate over the whole corpus, including the 85.4 percent in the published results file, is an artefact of a choice.

The mix is an assumption, and it decides the cost entirely. Section 04's three mixes are stated, bracketed and not measured. Nobody has published how often a real shipping desk legitimately introduces a new bank account or contact, and a deployment must measure its own before adopting this.

No attacker adapted. The obvious attack, routing payment to an account that is already on record, is untested and is the one I would expect to work.

The rest, in order of how much they bite

Fields, not prose. The check compares structured fields. An injection that changes what a draft *says* without changing a bank detail, recipient, link or attachment is invisible to it, and paper 13's style B contained exactly such requests.

No record staleness. The control is worth what the counterparty record is current. A stale record flags genuine changes until the reviewer learns to click through, which is worse than no control. Not measured, and probably the largest practical risk.

The 97.5 percent on new contacts is a corpus artefact. One generated "new" contact collided with an existing one, so the check correctly passed it. The real rate for a genuinely new address is 100 percent by construction.

Twelve counterparties. Enough to exercise the mechanism, nowhere near enough to represent a desk's record set, its duplicates, its stale rows or the same company under three spellings. Name normalisation is not tested and is where a real implementation would first break.

One language, one format. The drafts are structured records rather than rendered email. Nothing here tests extraction of a bank detail from HTML, a PDF attachment or a scanned invoice, which is how IBANs usually arrive.

Review quality is not measured. The paper stops at the flag. Whether a person shown a flagged draft makes the right decision, and how that degrades on the fortieth flag, is untested and is where the remaining risk sits.

No cross-check against paper 13's detectors on the same drafts. The comparison in section 03 is between this bench's drafts and paper 13's emails, which are different corpora. The claim that the detectors missed the bank account note is paper 13's result on paper 13's corpus, not a head-to-head on these 280 drafts. **That head-to-head has not been run** and it would be a stronger result than either paper has alone.

Conflict of interest and provenance

Unfunded. No sponsor, client or commissioning party paid for, commissioned, reviewed or approved this research, and no vendor was contacted.

The author builds AI systems commercially, including systems that read mail, and **this paper recommends a control he would be paid to implement.** That is the conflict, stated plainly. It is also why the check is published in forty lines of Python that anyone can read, run and criticise: the argument has to stand on the instrument rather than on the author, because the author is not disinterested.

Every counterparty, person, vessel, address, IBAN and figure in the corpus is invented, generated from a published script with a fixed seed. It describes no real organisation, correspondence, counterparty or client.

What would change the conclusion

Stated in advance, so that it cannot be quietly dropped later:

- A deployment measuring a legitimate novelty rate above about 50 per 1,000 would make the review load material and the control would need rethinking rather than adopting.
- An adaptive attacker reaching payment details through an on-record account would narrow the control's value to the other three field classes.
- Evidence that reviewers approve flagged drafts at a high rate would make the whole chain decorative, and that is the measurement I would most like somebody to run.

08 What follows

> Discharges claims J002, J008, J015, J027, J035, J037, J041 to J042, J044, J047, J053 to J054, J058 to J060.

If you run an AI assistant over a shipping mailbox

1. Build the counterparty record first. Bank details, contacts and link domains per counterparty, from whatever system already holds them, with a stated owner and a stated update path. **Without it none of this works**, and with it you have the asset the control is made of. This is the project; the check is forty lines.

2. Compare the draft, not the incoming email. Every control in paper 13's survey that judged text an attacker wrote was eventually beaten. A comparison against a record the attacker cannot write to has nothing to persuade.

3. Flag on four fields: bank details, recipients, link domains, attachments. 160 of 160 fraudulent drafts in this bench, at under two minutes of review a day on a hundred drafts, under stated assumptions about legitimate novelty.

4. Show the reviewer the raw draft and the comparison it failed, never the assistant's explanation of why the change is fine. That explanation is what the attacker wrote.

5. Measure your own legitimate novelty rate before you trust the cost figure. A few per thousand makes this free. Weekly bank changes make it a different conversation.

6. Watch the record going stale, because that is how this control dies. Flags on genuine changes train reviewers to click through, which manufactures the habit the attack needs.

If you are buying rather than building

Three questions that separate a vendor who has thought about this from one who has not:

- **Does your assistant have a send tool, and can it be removed?** If sending cannot be removed, the rest of the conversation is about reducing a rate rather than removing a capability.
- **What do you compare an outgoing draft against, and who maintains it?** "We detect suspicious content" is the answer that failed in paper 13.

- **What does the reviewer see when something is flagged?** If the answer is the assistant's

summary, the review is theatre.

What I would measure next

The head-to-head this paper lacks. Run paper 13's four detectors over these 280 drafts and this check over paper 13's 308 emails. Both papers claim the controls fail in different directions, and neither has demonstrated it on one corpus. That is a day of work and it would make both results stronger than they are apart.

The adaptive attack. Specifically, a fraudulent draft that routes payment to an account already on record, which is the hole I can see from here and have not tested.

Reviewer behaviour, which is the only part of the chain nobody in this literature measures and the part everything else depends on. How often is a flagged draft approved, and what happens to that rate by the fortieth flag on a Friday afternoon.

Extraction from real formats. This bench compares structured fields. Real bank details arrive in HTML, in PDFs and in scans, and getting them out reliably enough to compare is an unmeasured step sitting underneath the whole control.

The one line version

Paper 13 showed that no detector reliably spots the email asking an assistant to confirm a new bank account. This one shows that you do not need to spot it: **compare the draft against what you already know about the counterparty, and the request that defeats every classifier is caught every time, because the answer was never in the language.** What it costs is a record somebody keeps current, and what it does not do is decide anything, which is why a person still sends the mail.

ABOUT THE RECORD

A deterministic check of 280 synthetic outgoing drafts against twelve invented counterparties. No model is involved. Every counterparty, person, vessel, IBAN and figure is invented, and the generator, the check and every per-draft verdict are published with the paper.

WHAT IT SHOWS

The attack that defeats a classifier cannot defeat a comparison, because the comparison never reads the attacker text. What it costs is a counterparty record somebody keeps current, and what it does not do is decide anything.

LIMITS AND CONFLICTS

The class balance is a design parameter, so the aggregate rates in the results file are artefacts and the per-class rates are the measurement. No attacker adapted, and routing payment to an on-record account is untested. The author builds AI systems commercially and this paper recommends a control he would be paid to implement, which is why the check is published in full. Unfunded, and no vendor was contacted.

© 2026 Nikolaos Broikos. Text and data under CC BY 4.0, code under MIT. The data, code and claim ledger are in data/paper-14/ so any figure here can be recomputed, checked or disputed.