There’s this phrase used by some technologists: ‘epistemic security’. Epistemic security has to do with things like the spreading of misinformation on social media. I.e. if we spend enough time on social media reading unreliable information, how does that influence us? That’s a matter of epistemic security. Epistemic security is interesting because if you start a conversation about epistemic security at a party, they don’t invite you back.

Publishing is a simple process in principle.
So peer-review is an epistemic security system. Peer-review stops bad papers getting through—and, therefore, stops bad information getting into our brains. But, there’s a problem. Peer-review isn’t perfect and it was never intended to be. Security systems have holes... like Swiss cheese.

I used to edit journals and I always wanted to get the best referees on every paper. But the reality is that sometimes you need to hold the best reviewers in reserve so that they are available when needed. You also need to give inexperienced reviewers the opportunity to learn and that means accepting that the best standard of review isn’t always possible.
The Papermill Alarm is an early-warning system which alerts you to papers with signs of papermilling. It helps to fill-in those holes. When an editor sees an alert, they know this is one of those times when they need to call on their best reviewers. It’s an easy decision to make as an editor and it turns out that it’s an easy solution to the problem, too.

We don’t actually prescribe how editors use the output from the Papermill Alarm, but one thing we’ve learned: peer-review is effective against papermills. So applying the best standard of peer-review is an effective control.

- I’m yet to see a situation where the rejection rate for papers which receive ‘red’ alerts on a journal isn’t significantly higher than the rejection rate for other papers.
- We know that papermills often try to manipulate the peer-review process (or circumvent it entirely—e.g. by abusing special issue programmes). They wouldn’t do that if peer-review wasn’t a problem for them.

That’s nice, isn’t it? It turns out that the thing we have always been good at—peer-review—is already the best tool for the job. The Papermill Alarm raises the alarm so that editors know when to take care.

So that’s why we recommend using peer-review to control the problem. But it’s up to you how you use these insights. There might be a couple of things to think about here...

**Detect and Respond**

I like this quote from Bruce Schneier. It’s about dealing with novel computer hacks. But it applies to novel research fraud, too.

> “You can’t prevent. You can’t defend. All you can do is detect and respond”
It sounds defeatist at first, doesn’t it? But I don’t read it that way: it makes the road forward clear and the nice thing is that we can already do the detection part.

Data science is the process of extracting value from data. With the Papermill Alarm, we’ve been very lucky to have not only high quality public datasets to start off with like Crossref, PubMed and OpenAlex, but also to have received data from publishers which describes the problem uniquely from multiple perspectives. It’s been said before, but the key to solving this problem is cooperation and it’s by working together that we’ve been able to build this unique data resource.

When it comes to papermills, the ‘hack’ isn’t novel any more. We can detect the overwhelming majority of problem papers before peer-review. Furthermore, using new methods, we can identify the individuals running the mills as well.

So detection is done. The question now is, given this rare opportunity, what is the best way to respond?

**Testing the tools**

Recently, I made an estimate for the global rate of papermilling (reported in Nature News by Richard Van Noorden).

I did the work back in April 2023 using one of our analytics pipelines.

I already had some good validation of the method at that time, such as the Chinese Academy of Sciences Early Warning Lists. The ‘red’ alerts are engineered to have zero (or near-zero) false positives in testing. However, test data isn’t the same as real-world data, so we needed to confirm it was performing well in the wild.

What we are seeing here is a trend in which particular template titles and abstracts appear in the literature. We know that the same templates are used by papermills and how to characterise them.

I spent much of the last year combing through the predictions from this method. What I found was that I would see dozens of journals with no alerts at all, and then I would see one journal with dozens or even hundreds of alerts.
It appeared that these rare journals had been targeted and so I spot-checked them by picking out individual papers and manually looking for problems. Once the spot check had confirmed that the false-positive rate was low, I would move on to the next journal.

While this was happening, I would report the findings to publishers. That led to some very useful feedback. Sometimes the journals were already under investigation and we even predicted a few mass-retractions that way.

I’d also like to acknowledge valuable feedback gratefully received from a community of sleuths throughout this process including Guillaume Cabanac, Cyril Labbé, Alexander Magazinov, Elisabeth Bik, Smut Clyde, Fidelia, Sholto David, Kaveh Bazargan, Cheshire.

At the journal level, when the alert-rate is high, we always find something.

There was another thing that was important here. Some publishers had almost no signal at all. I’d talk to them and they would tell me that they didn’t get papermills submitting because they were tough on peer-review and they thought that tough peer-review put the mills off. (And, as I said above, our data analysis on peer-review seems to confirm that they were right.)
So we know the false-positive rate is low. The chart above gives us 1.7–1.8%, but we have some potential for overestimation (e.g. there is a small amount of missing data which we can’t see). So I’m comfortable saying that I think 1.5% is a good rough lower-bound for the rate of papermilling in 2022.

Most of our methods are intended to be run at the article-level. So for the estimate, I just used one method that I knew would work at scale.

At this point, there’s certainly scope for someone to do a better estimate. But I don’t think it’s possible that would yield a rate lower than 1.5%.