Chapter 25

Taking Stock

But: connecting the world meant that we also connected all the bad things and all the bad people, and now every social and political problem is expressed in software. We've had a horrible 'oh shit' moment of realisation, but we haven't remotely worked out what to do about it.
– Benedict Evans

If you campaign for liberty you're likely to find yourself drinking in bad company at the wrong end of the bar.
– Whit Diffie

25.1 Introduction

Our security group at Cambridge runs a blog, `www.lightbluetouchpaper.org`, where we discuss the latest hacks and cracks. Many of the attacks hinge on specific applications, as does much of the cool research. Not all applications are the same, though. If our blog software gets hacked it will just give a botnet one more server, but there are other apps from which money can be stolen, others that people rely on for privacy, others that mediate power, and others that can kill.

I've already discussed many apps from banking through alarms to prepayment meters. In this chapter I'm going to briefly describe four classes of application at the bleeding edge of security research. They are where we find innovative attacks, novel protection problems, and thorny policy issues. They are: autonomous and remotely-piloted vehicles; machine learning, from adversarial learning to more general issues of AI in society; privacy technologies; and finally, electronic elections. What these have in common is that while previously, security engineering was about managing complexity in technology with all its exploitable side-effects, we are now bumping up against complexity in human society. Autonomous cars are hard because of the people driving other cars on the same road. AI is hard because our cool new pattern-matching tools, such as deep neural networks, can pick out not just real patterns in human behaviour –

sometimes unexpected ones – but false ones too. Privacy is hard because of the richness of human interaction in society. And elections are hard not just because of the technical difficulty of counting votes in a way that preserves both privacy and auditability, but because of the huge variety of dirty tricks used by political players, both upstream and downstream of the voting process itself. All of these problems explore, in various ways, the boundary between what humans can do and what machines can do.

25.2 Autonomous and remotely-piloted vehicles

The aviation pioneer Lawrence Sperry invented the first autopilot in 1912 and demonstrated it in 1914, flying past the judges in a 'safer aircraft' competition in Paris with his hands up. In the process he and his father Elmer invented the artificial horizon. A fixed-wing aircraft left to its own devices will eventually go into a spiral dive and crash; the pilot can keep it level with reference to the horizon, but when flying in cloud that external reference is missing. A gyroscope can provide the missing reference, and it can also drive the ailerons and elevators via servos.

In 1975, I got my first proper job re-engineering a fast-jet inertial navigation set to work on the midget submarines used in the oil industry. Engineers in the same building were working on early head-up displays and satellite navigation equipment. Each of these pieces of equipment weighed about 20kg and cost £250,000 – about $3M in today's money. All three together left little change out of $10M and weighed as much as a person.

Now, in 2020, you have all three in your phone. Rather than three spinning mechanical gyros in a precision-engineered cage, your phone has a chip with MEMS accelerometers and gyros. It also has a GPS chip for satellite navigation and a Google or Apple Maps app to show you how to walk, cycle or drive to your destination. Over forty years, the cost has fallen by six orders of magnitude and the mass by four. This has driven rapid evolution of assistive technology on sea, air and land. Pioneering single-handed yachtsmen developed self-steering gear to cross oceans from the 1920s, to give them time to sleep, cook and repair sails; amateurs now have smarter autopilots for coastal cruising. Autonomous probes swim beneath the Antarctic ice to measure how quickly it's melting. The world's navies develop underwater mines, autonomous submersibles to find them, and much else.

25.2.1 Drones

In the air, early weapons such as the German V1 and V2 used twin-gyro autopilots, while the Cold War gave us the Tomahawk cruise missiles used to great effect in both Gulf Wars. In service since the early 1980s, these sneak under the enemy radar by flying close to the ground, and use terrain contour matching to update their inertial navigation. They were followed closely by a variety of unmanned aerial vehicles (UAVs), which saw their first large-scale use in the war between Israel and Syria in 1982; the Israeli Air Force used them for reconnaissance and as decoys, wiping out the Syrian air force with minimal losses. The best-known of the next generation of UAVs was the Predator. Initially designed as a reconnaissance vehicle, it could linger over a target area at medium altitude for many hours, and was adapted to carry Hellfire missiles to strike targets on the ground. In service from 1995–2018, it saw service in Iraq, Afghanistan, Libya and elsewhere. It was replaced by the larger, faster Reaper, which became a mainstay of the war in Syria against Islamic State. The world's armed forces now have a large range of UAVs, right down to small drones that soldiers carry in their rucksacks and use to see what's round the next corner.

Through the 20th century, enthusiasts built small radio-controlled model aircraft, but the FAA only issued its first commercial drone permit in 2006. In 2010, Parrot unveiled its AR Drone, a quadcopter that could be controlled by wifi from a smartphone, and in 2013 Amazon announced it was considering drones for delivery. Interest took off rapidly; within a couple of years our students were building drones and soon you could buy low-cost models in hobby shops. The main application in 2020 is aerial photography. There have been both insurgent and criminal uses, though, with drones used to deliver both drugs and mobile phones to prisoners, while insurgents have fitted drones with improvised explosive devices for use as weapons.

25.2.2 Self-driving cars

Most of the recent surge in interest though has been in self-driving cars and trucks. In 2004, faced with mounting combat losses to improvised explosive devices in Afghanistan and Iraq, DARPA decided to push the development of self-driving vehicles, and announced a competition with a million-dollar prize for whoever built one that could cross 149 miles of the Mojave desert the fastest. The prize went unclaimed as no vehicle finished the course, but the following year a team from Stanford led by the roboticist Sebastian Thrun collected the prize, now two million. His robot, Stanley, used machine learning and probabilistic reasoning to cope with terrain perception, collision avoidance, and stable vehicle control on slippery and rugged terrain [1887]. This built on robotics research going back to the 1980s, much of which DARPA had also funded. Their next challenge in 2007 moved from the desert to a simulated urban environment; competitors had to detect and avoid other vehicles, and obey the rules of the road. This bootstrapped a research community and the technology started to improve quickly.

Previously, carmakers had been steadily adding assistive technology, starting with antilock braking systems (ABS) in the last century and progressing through adaptive cruise control (ACC), which I described in section 23.4.1, automatic emergency braking (AEB) and lane keeping assist (LKA). The industry vision was that these would eventually come together into a full autopilot. Inspired by the DARPA challenges, Google hired Sebastian Thrun to lead Project Chauffeur in 2009 with a goal of building a fully self-driving car. This was announced in 2010, stimulating a market race involving both the tech and auto industries. Tesla was the first to field a product in 2014, when its 'Autopilot' software was launched as an over-the-air upgrade that could take control on the freeway or in stop-start traffic. There was already a hype cycle underway for machine learning, which I'll discuss in the next section, and self-driving cars hitched a ride. Tesla's Elon Musk was predicting full autonomy by 2018, and Google's Sergey Brin by 2017, before the Google car project was spun off as Waymo in 2016. People talked excitedly about low-cost robotaxis causing personal car ownership to be replaced by mobility-as-a-service; the arrival of Uber added a further competitor, and the hype scared even auto industry execs who should have known better into predicting that by the mid-2020s people wouldn't own their own cars any more. The hype cycle passed, as it always does. As I write in 2020, Waymo is operating a limited self-driving car service in a 50-square-mile area of Phoenix [871]. The service isn't available when it's raining, or in dust storms, and is monitored in real-time by humans at a control centre. It had been announced several times, but problems kept on forcing the company to put safety drivers back in the cars. So what's going on?

A large part of the answer is that other road users are unpredictable. Automation can deal with some of the resulting hazards: if the car in front brakes suddenly, a robot can react faster. Adaptive cruise control cuts driver fatigue and even cuts congestion once enough vehicles use it, as it damps the propagation of shock waves through traffic. But even here there are limits. When engineers extended the technology to automatic emergency braking, the inability to infer the intentions of other drivers became a limiting factor. Suppose for example you're driving on an open country road when the car in front indicates a turn and starts to slow down. You maintain speed as you expect it'll have left the road by the time you get there, and if not you'll just overtake. But the AEB might not understand this, so as you get too close to the turning car it activates, throwing you forward on your seat belt. Consumer tests of AEB systems in 2020 still show quite some variability, both in the false alarm rate and in the ability to stop the car in time when a pedestrian dummy is pulled across the road. Some systems restrict activation to city rather than highway speeds, and in 2020 all tend to be options available on more expensive models. AEB should be in all new cars in about 2022. Since 2016 insurers have been happy that it reduces the overall risk; I'll discuss safety assurance in section 28.4.1.

But each new assistive technology takes years to optimise and debug, and it's not straightforward to combine a dozen of them into an autopilot. The paper that Sebastian Thrun and his team wrote to describe Stanley gives a useful insight into the overall technology [1887]. There are several dozen programs interacting loosely, reflecting our understanding of how humans do such tasks; your subconscious looks at all sorts of things and brings hazards to your attention. Simultaneous processes in Stanley handled path planning, steering control and obstacle avoidance; this used laser rangefinders up to 22m, a colour camera beyond that, and a radar beyond that (which was not used in the race, as Stanley was given over 2000 waypoints for a predetermined course). Each of these systems had to solve many subproblems; the vision system, for example, had to adapt to changing light conditions and road colour. Stanley then had to be optimised via extensive testing, where the objective function was to maximise the mean distance between catastrophic failure (defined as the human safety driver taking over).

Combining the subsystems means compromises, and while the main vendors hold their design details secret, we're starting to learn about the optimisations and what goes wrong with them from accidents. For example, when a self-driving Uber killed Elaine Herzberg in Arizona in March 2018, it emerged at the NTSB inquiry that Elaine had been pushing a bicycle and the vision system flapped between identifying her as a pedestrian and as something else, but ultimately she was not recognised as a pedestrian because she was not on a crosswalk. AEB might have stopped the car but it had been turned off "to reduce the potential for erratic vehicle behavior" – in other words, because the false alarm rate was annoying [457]. Ultimately, Uber relied on the safety driver – who was unfortunately watching TV at the time[1].

Now we've known for decades that relying on humans to take over in an emergency takes time: a human has to react to an alarm, analyse the alarm display on the console, scan the environment, acquire situational awareness, get into the optical flow, and take effective control. Even in commercial aviation, it takes a flight crew about eight seconds to regain control properly after an autopilot failure. You cannot expect a safety driver in a car to do much better.

25.2.3 The levels and limits of automation

For such reasons, the Society of Automotive Engineers sets out five levels of automation:

1. Driver assistance – the software controls either steering or speed, and the human driver does the rest of the work;
2. Partial automation – the software controls both steering and speed in some modes but the human driver is responsible for monitoring the environment and assuming control at zero notice if the software gets confused;
3. Conditional automation – the software monitors the environment, and controls both steering and speed, but assumes the human can take over if it gets confused;
4. High automation – the software monitors the environment and drives the car, in some driving conditions, without assuming that a human can intervene. If it gets confused it stops at the side of the road;
5. Full automation – the software can do everything a human can.

So far, vehicles available on the mass market only have *advanced driver assistance systems* (ADAS), namely levels one and two, and insurers consider words like 'autonomous' and 'autopilot' to be dangerous as they cause customers to assume that a vehicle is operating at Level 4, which can lead to accidents. The Arizona crash can be seen as a car operating at Level 2 while the safety driver operated at Level 3. Level 4 often assumes a backup driver sitting in a control centre, overseeing several dozen 'autonomous' cars, but they won't have the bandwidth to understand a hazard as quickly as a safety driver on the spot. They don't feel the road noise and accelerations, they can't use their peripheral vision, and above all, they are not immersed in the optical flow field

[1] In fact, the very first fatal crash involving a Tesla on autopilot claimed the life of a driver who appeared to be watching a movie on his laptop when his car ran under a truck [1394].

that is critical to driving a car (or landing an aircraft) safely, as we discussed in section 3.2.1.

To what extent is Level 5 feasible at all, unless we invent artificial general intelligence? John Naughton remarked that a downtown delivery driver's job is pretty safe, as the work demands all sorts of judgment calls such as whether you can double-park or even block a narrow street for half a minute while you dash up to a doorway and drop a parcel, as the cars behind honk at you [1417]. Another hard case is the cluttered suburban street with cars parked either side, where you are forever negotiating who goes first with oncoming vehicles, using a wave, a nod or even just eye contact. Even the current Level 2 systems tend to have difficulty when turning across traffic because of their inability to do this tacit negotiation. They end up having to be much more cautious than a human driver and wait for a bigger gap, which annoys human drivers behind them. And if you've ever tried to ease a car through the hordes of students on bicycles in a college town like Cambridge, or any urban traffic in India, you know that dealing with human traffic complexity is hard in many other situations. Can your self-driving car even detect hand signals from police officers to stop, let alone cope with eight students carrying a bed, or with an Indian temple procession?

As of 2020, the Level 2 systems have lots of shortcomings. Tesla can't always detect stationary vehicles reliably; it uses vision, sonar and radar but no lidar. (One Tesla driver in North Carolina has been charged after running into the back of a stationary police car [1118].) The Range Rover can't always detect the boundary between a paved road and grass, but perhaps that wasn't a priority for a 4 x 4. Many cars have issues with little roundabouts, not to mention potholes and other rough surfaces; the first time I got a ride in one, my teeth were rattled as we went over speed bumps at almost 30mph. Roadworks play havoc with automatic lane-keeping systems, as old white lines that have been painted over can be shiny black and very prominent in some light conditions, leading cars to oscillate back and forth between old and new markings [632]. There's a huge amount of research on such technical topics, from better algorithms for multi-sensor data fusion though driving algorithms that can provide an explanation for their decisions, to getting cars to learn routes as they travel them, just like humans do. Tesla even has a 'shadow mode' for its autopilot; when it's not in use, it still tries to predict what the driver will do next, and records its mispredictions for later analysis. This has enabled Tesla to collect billions of miles of training data across a vast range of road and weather conditions.

I'll discuss safety assurance in section 28.4.1 but the state of play in 2020 is that while Tesla and NHTSA claimed that there are fewer crashes after a Tesla customer activates Autosteer, an independent lab claimed there were more. Now as I discussed in section 14.3.1, falling asleep at the wheel is a major cause of accidents, accounting for 20% of the UK total. These tend to be at the serious end of the spectrum; they account for about 30% of fatal accidents and half of fatal accidents on freeways. (That's why we have laws to limit commercial drivers' hours.) So we ought to be able to save lives with a system that keeps your car in lane on the freeway, brakes to avoid collisions, and brings it to a stop at the side of the road if you don't respond to chimes. Why is this not happening?

I suspect we'll need to disentangle at least three different factors: the risk thermostat, the system's affordances, and the expectations created by marketing. First, the risk thermostat is the mechanism whereby people adapt to a perceived reduction in risk by adopting more risky behaviour; we noted in section 3.2.5.7 that mandatory seat-belt laws caused people to drive faster, so that the overall effect was merely to move casualties from vehicle occupants to pedestrians and cyclists, rather than to reduce their number overall. Second, affordances condition how we interact with technology, as we discussed in section 3.2.1, and if a driver assistance system makes driving easier, and apparently safer, people will relax and assume it is safer – disposing some of them to take more risks. Third, the industry's marketing minimises the risks in subtle ways. For Tesla to call its autosteer feature an autopilot misled drivers to think they could watch TV or have a nap. That is not the case with an autopilot on an airplane, but most non-pilots don't understand that.

25.2.4 How to hack a self-driving car

The electronic security of road vehicles started out in the last century with the truck tachographs and speed limiters we discussed in section 14.3 and the remote key entry systems we discussed in section 4.3.1. It has become a specialist discipline since about 2005, when the carmakers and tier-1 component vendors started to hire experts. By 2008, people were working on tamper resistance for engine control units: the industry had started using software to control engine power output, so whether your car had 120 horsepower or 150 was down to a software switch which people naturally tried to hack. The makers tried to stop them. They claimed they were concerned about the environmental impact of improperly tuned cars, but if you believe that, I have a bridge I'd like to sell you.

In 2010, Karl Koscher and colleagues got the attention of academics by showing how to hack a late-model Ford. Cars' internal data communications use a CAN bus which does not have strong authentication, so an attacker who gets control of (say) the radio can escalate this access to operate the door locks and the brakes [1085]. In 2015, Charlie Miller and Chris Valasek got the attention of the press when they hacked a Jeep Cherokee containing a volunteer journalist, over its mobile phone link, slowed the vehicle down and drove it off the road [1316]. This compelled Chrysler to recall 1.4m vehicles for a software patch, costing the company over $1bn. This finally got the industry's attention.

There's now a diverse community of people who hack cars and other vehicles. There are hobbyists who want to tune their cars; there are garages who also want to use third-party components and services; and there are farmers who want to repair their tractors despite John Deere's service monopoly, as I mentioned in section 24.6. There are open-source software activists and safety advocates who believe we're all safer if everything is documented [1792]. And there are the black hats too: intelligence agencies that want to spy on vehicle occupants and thieves who just want to steal cars.

Car theft is currently the main threat model, and we discussed the methods used to defeat remote key entry and alarm systems in section 4.3.1. State actors and others can take over the mobile phones embedded in cars, using the techniques discussed in section 2.2.1. The phones, navigation and infotainment systems are often poorly designed anyway – when you rent a car, or buy one secondhand, you often see a previous user's personal information, and we described in section 22.3.3 how an app that enables you to track and unlock a rental car let you continue to do this once the car had been rented to somebody else.

So what else might go wrong, especially as cars become more autonomous? A reasonable worst-case scenario might see a state actor, or perhaps an environmental activist group, trying to scare the public by causing thousands of simultaneous road traffic accidents. A remote exploit such as that on the Chrysler Jeep might already do this. The CAN bus which most modern cars use for internal data communications trusts all its nodes. If one of them is subverted it might be reprogrammed to transmit continuously; such a 'blethering idiot', as it's called, makes the whole bus unusable. If this is the powertrain bus, the car becomes almost undriveable; the driver will still have some steering control but without power assistance to either steering or brakes. If the car is travelling at speed, there's a serious accident risk. The possibility that a malicious actor could hack millions of cars causing tens of thousands of road traffic accidents simultaneously is unacceptable, and such vulnerabilities therefore have to be patched. But patching is expensive. The average car might contain 50–100 electronic control units from 20 different vendors, and the integration testing needed to get them to all work together smoothly is expensive. I'll discuss this in more detail in section 27.5.4.

Attacks are not limited to the cars themselves. In 2017, Elon Musk told an audience, "In principle, if someone was able to say hack all the autonomous Teslas, they could say – I mean just as a prank – they could say 'send them all to Rhode Island' – across the United States ... and that would be the end of Tesla and there would be a lot of angry people in Rhode Island.". His audience laughed, and three years later it emerged that he'd not been entirely joking. A few months previously, a hacker had gained control of the Tesla 'mothership' server which controls its entire fleet; luckily he was a white hat and reported the hack to Tesla [1119]. At the other end of the scale, the performance artist Simon Weckert pulled a handcart containing 99 Android phones around Berlin in February 2020, causing Google Maps to register a traffic jam wherever he went [1997]. As advanced driver assistance systems rely ever more extensively on cloud facilities, the scope for such indirect attacks will increase.

And external attacks need not involve computers. If car systems start to slow down automatically for pedestrians and cyclists, some of them may exploit this. In India and some parts of southern Europe, pedestrians walk through congested traffic, flagging cars to stop, and they do; it will be interesting to see if this behaviour appears in London and New York as well.

Companies will exploit assistance systems if they can. Now that the initial dream of self-driving trucks seems some way off, and even the intermediate dream of multiple trucks driving in convoy between distribution hubs with a single driver seems ambitious, may we expect lobbying to relax the legal limits on drivers' hours? Trucking firms may argue that once the truck's on autopilot on the freeway, the driver only has to do real work on arrival and departure, so he should work ten hours a shift rather than eight. But if the net effect of the technology is to make truck drivers work more time for the same money, it will be resented and perhaps sabotaged.

Should Level 5 automation ever happen, even in restricted environments – so that we finally see the robotaxis Google hoped to invent – then we'll have to think about social hacking as a facet of safety. If your 12-year-old daughter calls a cab to get a ride home from school, then at present we have safeguards in the form of laws requiring taxi drivers to have background checks for criminal records. Uber tried to avoid these laws, claiming it wasn't a taxi company but a 'platform'; in London, the mayor had to ban them and fight them in court for years to get them to comply. So how will safeguarding work with robotaxis?

There will also be liability games. At present, car companies try to blame drivers for crashes, so each crash becomes a question of which driver was negligent. If the computer was driving the car, though, that's product liability, and the manufacturer has to pay. There have been some interesting tussles around the safety figures for assisted driving, and specifically whether the carmakers undercount crashes with autopilot activated, which we'll discuss in section 28.4.1.

So much is entirely predictable. But what about new attacks on the AI components of the systems themselves? For example, can you confuse a car by projecting a deceptive image on a bridge, or on the road, and cause it to crash? That's quite possible, and I've already seen a crash caused by visual confusion. On the road home from my lab, there was a house at a right-hand bend whose owner often parked his car facing oncoming traffic. At night, in a left-hand driving country like Britain, your driving reflex is to steer to the left of the facing car, but then you'd notice you were heading for his garden wall, and swerve right to pass to the right of his car instead. Eventually a large truck didn't swerve in time, and ended up in the wall.

So could clever software fool a machine vision system in new ways, or ways that might be easier for an attacker to scale? That brings us to the next topic, artificial intelligence, or to be more precise, machine learning.

25.3 AI / ML

The phrase *artificial intelligence* has meant different things at different times. For pioneers like Alan Turing, it ranged from the Turing test to attempts to teach a computer to play chess. By the 1960s it meant text processing, from Eliza to early machine translation, and programming in Lisp. In the 1980s there was a surge of research spurred by Japan's announcement of a huge research programme into 'Fifth generation computing', with which Western nations scrambled to keep up; much of that effort went into rule-based systems, and Prolog joined Lisp as one of the languages on the computer science curriculum.

From the 1990s, the emphasis changed from handcrafted systems with lots of rules to systems that learn from examples, now called *machine learning* (ML). Early mechanisms included logistic regressions, support vector machines (SVMs) and Bayesian classifiers; progress was driven by applications such as natural language processing (NLP) and search. While the NLP community developed custom methods, the typical approach to designing a payment fraud detector or spam filter was to collect large amounts of training data, write custom code to extract a number of signals, and just see empirically which type of classifier worked best on them. Search became intensely adversarial during the 2000s as search engine optimisation firms used all sorts of tricks to manipulate the signals on which search engines rely, and the engines fought back in turn, penalising or banning sites that use underhand tricks such as hidden text. Bing was an early user of ML, but Google avoided it for years; the engineer who ran search from 2000 until he retired in 2016, Amit Singhal, felt it was too hard to find out, for a given set of results, exactly which of the many inputs was most responsible for which result. This made it hard to debug machine-learning based algorithms for search ranking. If you detected a botnet clicking on restaurants in Istanbul and wanted to tweak the algorithm to exclude them, it was easier to change a few 'if' statements than retrain a classifier [1300].

A sea change started in 2011 when Dan Cireşan, Ueli Meier, Jonathan Masci and Jürgen Schmidhuber trained a deep convolutional neural network to do as well as humans on recognising handwritten digits and Chinese characters, and better than humans on traffic signs [435]. The following year, Alex Krizhevsky, Ilya Sutskever and Geoff Hinton used a similar *deep neural network* (DNN) to get record-breaking results at classifying 1.2 million images [1098]. The race was on, other researchers piled in, and 'deep learning' started to get serious traction at a variety of tasks. The most spectacular result came in 2016 when David Silver and colleagues at Google Deepmind produced AlphaGo, which defeated the world Go champion Lee Sedol [1737]. This got the attention of the world. Before then, few research students wanted to study machine learning; since then few want to study anything else. Undergraduates even pay attention in classes on probability and statistics, which were previously seen as a chore.

25.3.1 ML and security

The interaction between machine learning and security goes back to the mid-1990s. Malware writers started using tricks such as polymorphism to evade the classifiers in anti-virus software, as I described in section 21.3.5; banks and credit card companies started using machine learning to detect payment fraud, as I described in section 12.5.4; and phone companies also used it for first-generation mobiles, as I noted in section 22.2. The arrival of spam as the Internet opened up to the public in the mid-1990s created a market for spam filters. Hand-crafted rules didn't scale well enough for large mail service providers, especially once botnets appeared and spam became the majority of email, so spam filtering became a big application.

Alice Hutchings, Sergo Pastrana and Richard Clayton surveyed the use of machine-learning in such systems, and the tricks the bad guys have worked out to dupe them [939]. As spam filtering takes user feedback as its ground truth, spammers learned to send spam to accounts they control at the big webmail firms, and mark it 'not spam'; other statistical analysis mechanisms are now used to detect this. Poisoning a classifier's training data is a quite general attack. Another is to look for weak points in a value chain: airline ticket fraudsters buy an innocuous ticket, pass the fraud checks, and then change it just before departure to a ticket to a high-risk destination. And there are vigorous discussions of such techniques on the underground forums where the bad actors trade not just services but boasts and tips. Battista Biggio and Fabio Rolli give more technical background: in 2004, spammers found they could confuse the early linear classifiers in spam filters by varying some of the words, and an arms race took off from there [241].

It turns out that these attack ideas generalise to other systems, and there are other attacks too.

25.3.2 Attacks on ML systems

There are at least four types of attack on a machine-learning system.

First, you can poison the training data. If the model continues to train itself in use, then it might be simple to lead it astray. Tay was a chatbot released by Microsoft in March 2016 on Twitter; trolls immediately started teaching it to use racist and offensive language, and it was shut down after only 16 hours.

Second, you can attack the model's integrity in its inference phase, for example by causing it to give the wrong answer. In 2013, Christian Szegedy and colleagues found that the deep neural networks which had been found to classify images so well in 2012 were vulnerable to *adversarial samples* – images perturbed very slightly would be wildly misclassified [1857]. The idea is to choose a perturbation that maximises the model's prediction error. It turns out that neural networks have plenty of such blind spots, which are related to the training data in non-obvious ways. The decision space is high-dimensional, which makes blind spots mathematically inevitable [1706]; and with neural networks the decision boundaries are convoluted, making them non-obvious. Researchers quickly came up with real-world adversarial examples, ranging from small stickers that would cause a car vision system to misread a 30mph speed sign as 60mph, to coloured spectacles that would cause a man wearing them to be misrecognised as a woman, or not recognised at all [1720]. In the world of malware detection, people found that non-linear classifiers such as SVM and deep neural networks were not actually harder to evade than linear classifiers provided you did it right [241].

Third, Florian Tramèr and colleagues showed that you can attack the model's confidentiality in the inference phase, by getting it to classify a number of probe inputs and building a successively better approximation. The result is often a good working imitation of the target model. As in the manufacture of real goods, a knock-off is often cheaper; big models can cost a lot to train from scratch. This approximation attack works not just with neural networks but also with other classifiers such as logistic regression and decision trees [1901].

What's more, many attacks turn out to be transferable, so an attacker doesn't need full access to the model (a so-called *white-box attack*) [1900]. Many attacks can be developed on one model and then launched against another that's been trained on the same data, or even just similar data (a *black-box attack*). The blind spots are a function of the training data, so in order to make attacks less transferable you have to make an effort. For example, Ilia Shumailov, Yiren Zhao, Robert Mullins and I have experimented with inserting keys in neural networks so that the blind spots appear in different places, and models with different keys are vulnerable to different adversarial samples [1733]. Kerckhoffs' principle applies in machine learning, as almost everywhere else in security.

A variant on the confidentiality attack is to extract sensitive training data. Large neural networks contain a lot of state, and the simplest way to deal with outliers is often just to memorise them. So if some business claims that a classifier trained on a million medical records is not personal data because it's "statistical machine learning", take care. Ways of combining machine learning with differential privacy, which we discussed in section 11.3, are a subject of active research [1493].

Finally, you can deny service, and one way is to choose samples that will cause the classifier to take as long as possible. Ilia Shumailov and colleagues found that one can often deny service by posing a conundrum to a classifier. Given a straight-through pipeline, as in a typical image-processing task, a confusing image can take 20% more time, but in more complex tasks such as natural language processing you can invoke exception handling and slow things down hundreds of times [1730].

More complex attacks straddle these categories. For example, there's an arms race between online advertisers and the suppliers of ad-blocking software, and as the advertisers adopt ever more complicated ways of rendering web pages to confuse the blockers, the blockers are starting to use image processing techniques on the rendered page to spot ads. However this leaves them open to advertisers using adversarial samples either to escape the filter, or to cause it to wrongly block another part of the page [1899].

So how can one use machine learning safely in the real world? That's something we're still learning, but there are some things we can say. First, one has to take a systems security approach and look at the problem end-to-end. Just as we sanitise inputs to web services, do penetration testing, and have mechanisms for responsible disclosure and update, we need to do the same for ML systems [659].

Second, we need to draw on the experience of the last twenty years' work on topics like card fraud, spam and intrusion detection. As we mentioned in section 21.4.2.2, ML systems have been largely ineffective at real-world network intrusion detection; Robin Sommer and Vern Paxson were the first to give a good explanation why. They discuss the lack of training data, the distance between theory and practice, the difficulties in evaluation, the high cost of errors and above all the inability to deal with novel attacks [1802]. The problem of keeping capable opponents out of complex corporate networks just isn't one that artificial intelligence has ever been good at.

There may occasionally be a change in emphasis, though. If we want to lower the probability of a new adversarial attack causing real damage, there are various things we can do, depending on the context. One is simply to detune the classifier; this is the approach in at least one machine-vision system used in cars. By making it less sensitive, you make it less easy to spoof, and then you complement it with other sensors such as radar and ultrasonics so that the vision system on its own is less critical. An alternative approach is to head in the other direction, by making the ML component of your system sufficiently fragile that an attack can be detected by other components – whereupon you switch to a defensive mode of operation, such as a low-sensitivity limp-home mode or stopping and waiting for a human to drive. In other words, you set out to build in situational awareness. This is how we behave in real life; as I discussed in section 3.2.5.1, the ancestral evolutionary environment taught us to take extra care when we sense triggers such as adversarial intent and violations of tribal taboos. So we've experimented with using neural networks trained so that a number of outputs and activations are considered to be taboo and avoided; if any of these taboos is broken, an attack can be suspected [1733].

The fundamental problem is that once we start letting machine learning blur the boundary between code and data, and systems become data-driven, people are going to game them. This brings us to the thorny problem of the interaction of machine learning and society.

25.3.3 ML and society

The surge of interest in machine learning since 2016, and its representation as 'artificial intelligence' in the popular press, has led to a lot of speculation about ethics. For example, the philosopher Dan Dennett objects on moral grounds to the existence of persons that are immortal and intelligent but not conscious. But companies already meet that definition! The history of corporate wrongdoing shows that corporations can behave very badly indeed (we discussed some examples in section 12.2.6). The most powerful ML systems belong to corporations such as Google, Amazon, Microsoft and IBM, all of which have had tussles with authority. The interplay between ML, big data and monopoly adds to the thicket of issues that governments need to navigate as they ponder how to regulate tech. One aspect is that the tech majors' ML offerings are now becoming platforms on their own, and used by lots of startups solving specific real-world problems [658].

One cross-cutting issue is prejudice. Aylin Caliskan, a Turkish research student at Princeton, noticed that machine translations from Turkish to English came out with gender bias; although Turkish has no grammatical gender, the English translations of Turkish sentences would assign doctors as 'he' and nurses as 'she'. On further investigation, she and her supervisors Joanna Bryson and Arvind Narayanan found that essentially all machine translation systems in use were not merely sexist, but racist and homophobic too [369]. In fact a large number of natural-language systems based on machine learning inhale the prejudices of their training data. If the big platforms' ML engines can suffuse prejudice through the systems on which hundreds of downstream firms rely, there is definitely a public-policy issue.

A related policy problem is *redlining.* When insurance companies used postcode-level claim statistics to decide the level of premiums, it was found that many minority areas suffered high premiums or were excluded from cover, breaking anti-discrimination laws. I wrote in the second edition of this book in 2008: "If you build an intrusion detection system based on data mining techniques, you are at serious risk of discriminating. If you use neural network techniques, you'll have no way of explaining to a court what the rules underlying your decisions are, so defending yourself could be hard. Opaque rules can also contravene European data protection law, which entitles citizens to know the algorithms used to process their personal data."

A second cross-cutting issue is snake oil, and the AI/ML gold rush has led to thousands of startups, many of them stronger on marketing than on product. Manish Raghavan and colleagues surveyed 'AI' systems used in employment screening and hiring, finding dozens of firms that claim their systems match new hires to the company's requirements. Most claim they don't discriminate, yet as few employers retain comprehensive and accessible data on employee performance, it's entirely unclear how such systems can even be trained, let alone how a firm that used such a system might defend a lawsuit for discrimination [1571]. Applicants quickly learn to game the system, such as by slipping the word 'Oxford' or 'Cambridge' into their CV in white text. A prudent employer would demand more transparent mechanisms, and devise independent metrics to validate their outcomes. Even that is nontrivial, as machine learning can discover correlations that we do not understand.

Arvind Narayanan has an interesting analysis of snake oil in AI [1382]. 'AI' and even 'ML' are generic terms for a whole grab-bag of technologies. Some of them have made real progress, like DNNs for face recognition, and indeed AlphaGo. So companies exploit this hype, slapping the 'AI' label on whatever they're selling, even if its mechanisms use statistical techniques from a century ago. Digging deeper, Arvind argues that machine-learning systems can be sorted into three categories:

1. ML has made real progress on tasks of *perception*, such as face recognition (see section 17.3), the recognition of songs by products like Shazam (see section 24.4.3), medical diagnosis from scans, and speech-to-text – at all of which it has acquired the competence of skilled humans;
2. ML has made some progress on tasks of *judgment*, such as content recommendation and the recognition of spam and hate speech. These have many hard edge cases about which even skilled humans disagree. The systems that perform them often rely on substantial human input – from a billion email users clicking the 'report spam' button to the tens of thousands of content moderators employed by the big tech companies;
3. ML has made no progress on tasks of *social prediction*, such as predicting employee performance, school outcomes and future criminal behaviour. A very extensive study by Matthew Sagalnik and over 400 collaborators has concluded that insofar as life outcomes can be predicted at all, this can be done as well using simple linear regressions based on a handful of variables [1638].

This is a falsifiable claim, so we'll see how accurate it is over time, and if there's a fourth edition of this book in 2030 we'll have a lot more data then. A major theme of research meanwhile will be to look for better ways for people and machines to work together. Intuitively, we want people to do the jobs involving judgment and machines to do the boring stuff; but making that actually work can be harder than it looks. Often people end up being the machine's servants, and according to one VC firm, 40% of 'AI' startups don't actually use ML in any material way; they're merely riding the wave of hype and employ people behind the scenes [1960]. One way or another, there will be lots of bumps in the road, and lots of debates about ethics and politics.

Perhaps the best way to approach the ethics is this. Many of the problems now being discussed in the context of AI ethics arose years ago for research done using traditional statistical methods on databases of personal information. (Indeed, linear regressions have been used continuously for about a century; they've just been rebranded as machine learning.) So our first port of call should be existing law and policy. When we discussed ethics in the context of records-based health and social-policy research in section 10.4.5.1, we observed that many of the issues arose because IT companies and their customers ignored the wisdom that doctors, teachers and others had accumulated over years of dealing with paper-based records. The same mistakes are now being repeated, and excused as before with sales hype around 'innovation' and 'disruption'.

In the case of predicting which children are likely to turn to crime, it's been known for years that such indicators can be deeply stigmatising. In section 10.4.6 we noted that if you tell teachers which kids have had contact with social services, then the teachers will have lower expectations of them. Both child welfare and privacy law argue against sharing such indicators. How much more harmful might it be if clueless administrators buy software that claims to be making predictions using the inscrutable magic that enabled AlphaGo to beat Lee Sedol? As for 'predictive policing', studies suggest that it might just be another way to get the computer to justify a policy of 'round up the usual suspects' [677]. (In section 14.4 we discussed how curfew tags also have this effect.) Similar issues arise with the use of ML techniques to advise judges in bail hearings about whether a suspect poses a flight risk or reoffending risk, and also in sentencing hearings about whether a suspect is dangerous. Such technologies are likely to propagate existing social biases and power structures, and provide lawmakers with an excuse to continue ineffective but populist policies, rather than nudging them to tackle the underlying problems.

ML is nonetheless likely to upset some of the equilibria that have emerged over the years on issues like surveillance, privacy and censorship, as it makes even more powerful tools available to already powerful actors, as well as creating new excuses to revive old abuses. Many countries already restrict the use of CCTV cameras; now that face-recognition systems enable pedestrians to be recognised, do we need to restrict them more? As we saw in section 17.3, a number of cities (including San Francisco) have decided the answer is 'yes'. In section 11.2.5 we discussed how location and social data can now make it very hard to be anonymous, and how people's Facebook data could be mined for political ad targeting. ML techniques make it easier to do traffic analysis, by spotting patterns of communication [1719]; in fact, police and intelligence agencies depend ever more on traffic and social-network analysis of the sort discussed in sections 21.7, 23.3.1 and 26.2.2.

In short, the charge sheet against machine learning is that it is one of the technologies helping entrench the power of the tech majors while pushing the balance between privacy and surveillance towards surveillance and facilitating authoritarian government in other ways. It may be telling that Google and Microsoft are funding big research programs to develop AI for social good.

So what can we do as a practical matter to get some privacy in this electronic village in which we now live?

25.4 PETS and operational security

Even if you don't blurt out all your thoughts on Facebook, social structure – who hangs out with whom – says an awful lot, and has become much more visible. In section 11.2.5 we discussed research which suggested that as few as four Facebook likes enable a careful observer to work out whether you're straight or gay most of the time, and how this observation led among other things to the Cambridge Analytica scandal, where voters' preferences were documented covertly and in detail.

Even if you don't use Facebook at all, the traffic data on who contacted whom gives a lot away to those who have access to it, as we discussed in section 11.4.1. This can cause problems for people who are in conflict with authority, such as whistleblowers. Anonymity can sometimes be a useful tool here. The abuse of academic authority is countered by anonymous student feedback on professors and anonymous refereeing of conference paper submissions. If your employer pays your health insurance, you might want to buy an HIV test kit for cash and get the results anonymously online, as the mere fact that you took a test says something, even if the result is negative. Privacy can also be a necessary precursor of free speech. People trying to innovate in politics or religion may need to develop their doctrine and build their numbers before going public. And then there are opposition politicians digging a bear trap for the government of the day, whose concerns are more tactical.

The importance of such activities to an open society is such that we consider privacy and freedom of speech to be interlinked human rights. We also enact laws to protect whistleblowers. But how can this work out in practice?

In pre-technological societies, two people could walk a short distance away from everyone else and have a conversation that left no hard evidence of what was said. If Alice claimed that Bob had criticised the king, then Bob could always claim the converse – that it was Alice who'd proposed a demonstration to increase the powers of parliament and he who'd refused out of loyalty.

In other words, many communications were *deniable*. Plausible deniability remains an important feature of some communications today, from everyday life up to the highest reaches of intelligence and diplomacy. It can sometimes be fixed by convention: for example, a litigant in England can write a letter marked 'without prejudice' to another proposing a settlement, and this letter cannot be used in evidence. But most circumstances lack such clear and convenient rules, and the electronic nature of communication often means that 'just stepping outside for a minute' isn't an option. What then?

A related issue is anonymity. Until the industrial revolution, most people lived in small villages, and it was a relief – in fact a revolution – to move into a town. You could change your religion, or vote for a land-reform candidate, without your landlord throwing you off your farm. In a number of ways, the effect of the Internet has been to take us back to an 'electronic village': electronic communications have not only shrunk distance, but in some ways our freedom too.

Can technology help? To make things a bit more concrete, let's consider some people with specific privacy problems.

1. Andrew is a missionary in Texas whose website has attracted a number of converts in Iran. That country executes Muslim citizens who change their religion. He suspects that some of the people who've contacted him aren't real converts, but religious policemen hunting for apostates. He can't tell a policeman apart from a real convert. What sort of technology should he use to communicate privately with converts?

2. Bella is your ten-year-old daughter, who's been warned by her teacher to remain anonymous online. What sort of training should you give her?

3. Charles is a psychoanalyst who sees private patients suffering from depression, anxiety and other problems. Previously he practised in a nondescript house in town which his patients could visit discreetly. Since lockdown, he's had to use tools like Skype and Zoom. What's prudent practice to protect patient privacy?

4. Dai is a human-rights worker in Vietnam, in contact with people trying to set up independent trade unions, microfinance cooperatives and the like. The police harass her frequently. How should she communicate with colleagues?

5. Elizabeth works as an analyst for an investment bank that's advising on a merger. She wants ways of investigating a takeover target without letting the target get wind of her interest – or even learn that anybody at all is interested. Her opponents are people like her at other firms.

6. Firoz is a gay man who lives in Tehran, where being gay is a capital offence. He'd like some way to download porn and perhaps contact other gay men without getting hanged.

7. Graziano is a magistrate in Palermo setting up a hotline to let people tip off the authorities about Mafia activity. He knows that some of the cops who staff the office in future will be in the Mafia's pay – and that potential informants know this too. How does he limit the damage that corrupt cops can do?

8. Hristo helps refugees enter the UK so they can claim asylum. Most of his clients are fleeing wars or bad government in the Middle East and North Africa. He operates from Belgium and gets clients into trucks or on to speedboats depending on the weather. He needs to coordinate with colleagues in France, Britain and elsewhere. How can they do this despite surveillance from assorted security and intelligence agencies?

9. Irene is an investigative journalist on a combative newspaper who invites whistleblowers to contact her. She dreams of landing the next Ed Snowden. What preparations should she make in case she does get contacted by a major source that the government would try hard to unmask?

10. Justin is running for elected office. Irene would happily dig the dirt on his family; and there are many other people who want to read his email, send a racist tweet from his social media account, or wire his campaign war chest to North Korea. How can he frustrate them?