THE RISE OF MACHINE LEARNING IN CYBERSECURITY

As it becomes painfully clear that traditional cybersecurity solutions leave gaps, however small, that can be successfully exploited by adversaries, many users take heart when new and promising security solutions emerge. Technologies such as heuristics, deep packet inspection or behavioral analysis have brought hope of better protection in their time. Today, the latest trend in cybersecurity is artificial intelligence, and specifically machine learning (ML). The latter has been touted as the new remedy to security issues. However, a major challenge with ML is that due to its complexity, it's difficult for security professionals to truly evaluate the use and effectiveness of ML technology in security products. As stated by Dan Ariely, the James B. Duke Professor of Psychology and Behavioral Economics at Duke University's Fuqua School of Business, "Everyone talks about it, (but) nobody really knows how to do it. Everyone thinks everyone else is doing it, so everyone claims they are doing it."

The purpose of this white paper is to help users understand how CrowdStrike ® uses ML to protect endpoints. To get there, we must first clarify what ML is and how it works. Then we will describe how Crowdstrike implements ML, specifically in the area of malware detection. Finally, we will discuss the benefits and limitations of applying ML in cybersecurity. In the end, the reader will get a better understanding of ML and how — when used correctly — it can help defend against cyber threats.

"EVERYONE CLAIMS THEY ARE DOING IT."

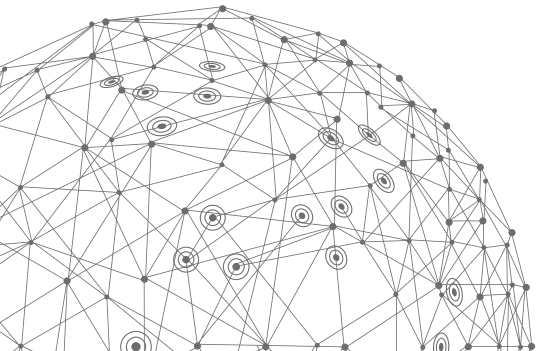
- **DAN ARIELY**, Duke University Professor of Psychology and Behavioral Economics

CHAPTER 1

WHAT IS MACHINE LEARNING?

It's been a few years since John Hennessy, past president of Stanford University, declared that "Machine Learning is the hot new thing." In fact, it's taken 60 years for ML technology to achieve that status. A similar concept, artificial intelligence (AI), seems to be following the same path and often, the terms AI and ML are used interchangeably . In reality, ML is a subset of AI, which covers a broad area of data analysis that enables algorithms to make decisions on their own by learning from data.

Data scientists have made huge progress since ML's humble debut in the 1950s, when it took a room full of computers to teach a machine how to play checkers. Today, ML has permeated our everyday lives so deeply that we commonly use it without even knowing it: every time we receive movie recommendations or shopping suggestions, for example, or when a credit card company alerts us of a potential fraud.

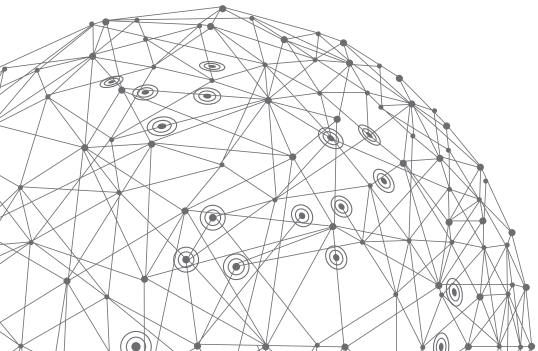


A Quick Definition

Machine learning is a subset of the broader field of artificial intelligence (AI). ML teaches a machine how to answer a question or how to make a decision on its own. It contrasts with traditional programming, which requires giving a machine explicit instructions for it to answer specific questions. In fact, every imaginable case has to be programmed ahead of time in order to cover all possible situations.

For example, imagine you wanted to take a multiple choice test. You could memorize all the correct answers by heart, which would be the equivalent of traditional programming, or you could learn to understand the concepts behind the questions, and then use that knowledge to determine the correct answer. The latter method represents the fundamental principle of ML.

The important difference is that ML teaches a machine how to predict an answer. This offers many advantages, but the biggest is the ability for the machine to respond to situations that it has not specifically encountered before, replacing processes that would have required arduous and time-intensive human analysis.



A Vast Field

In a nutshell, ML learns by being fed multiple examples in the form of a dataset, and rules or algorithms to apply to that dataset. The more examples the machine sees, the better it can learn.

There are multiple types of ML and each works very differently. If we generalize the field, we can define three main categories of ML: supervised learning, unsupervised learning and reinforcement learning.

> Supervised Learning

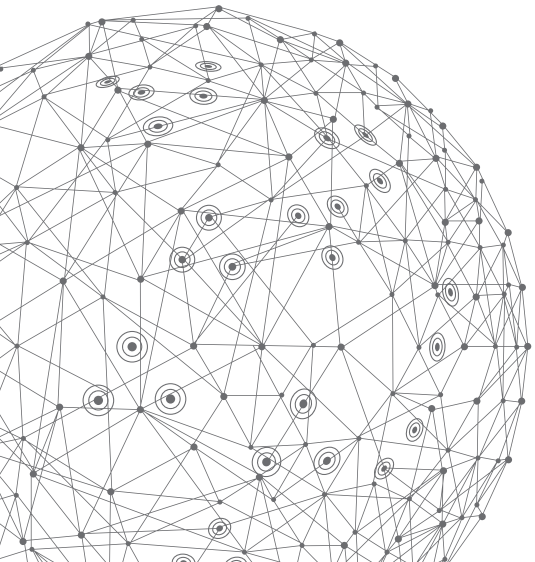
In supervised learning, the machine is trained using sample data that is labeled to tell the machine what the data represents. In other words, it knows what it's looking at (the input) and it knows what answers are expected (the output). Based on that training, the machine should be able to analyze new data and predict the correct answer. Supervised learning has applications such as disease diagnostics, or speech recognition.

> Unsupervised Learning

In unsupervised learning, the machine is trained using data that doesn't have labels. That means that the machine does not know what the data represents nor what answers are expected. The machine will have to figure out on its own the patterns and structure of the unlabeled input and discover the expected output. The classification of movie genres in Netflix is an example of unsupervised learning.

> Reinforcement Learning

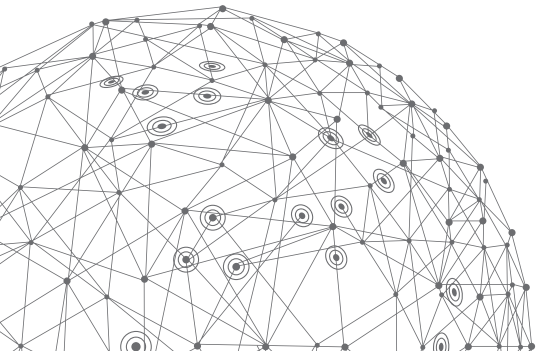
In reinforcement learning, the machine interacts with its environment to achieve a certain goal. It is similar to unsupervised learning, as the machine is trained using unlabeled data. However, in reinforcement learning, the machine receives feedback on the outcome. For example, a machine can use this model to learn how to play a game. If the machine receives positive feedback (it wins) or negative feedback (it loses) from the actions it takes, it will, over time, determine by itself the best strategy to win the game. Each victory will reinforce the validity of specific actions. Reinforcement learning applications are emerging in robotics for manufacturing.

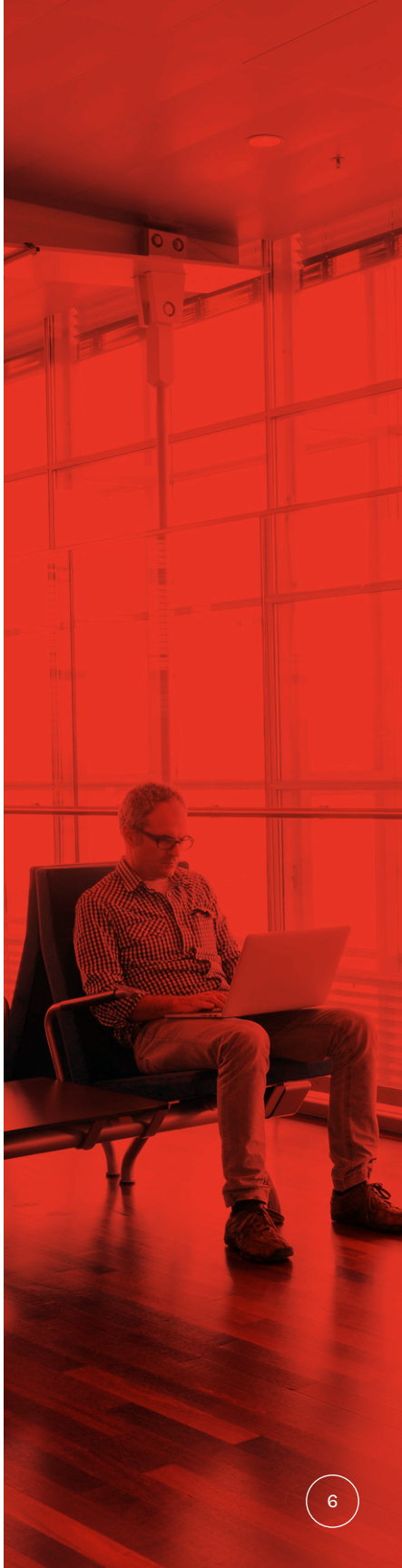


CHAPTER 2

HOW MACHINE LEARNING WORKS

In this white paper, we are going to focus on supervised ML, as it is well-suited for malware detection. In order to understand ML concepts, we will show an example of how to build a supervised ML model.





FINDING ENOUGH OF THE RIGHT DATA

Label	Man	Woman	Woman	Man	Man
Height	1800	1720	1630	1850	1750
Arm length	965	870	890	937	908
Weight	80	59	65	68	78
Finger length	100	80	79	90	85
Backside circumference	1003	981	789	807	998
Foot size	420	390	380	400	396

Supervised ML needs examples to learn. This means that to get started, we need to find a relevant dataset that we can use to train the model. Not only do we need to find appropriate data, but we need to find enough of it to train the machine.

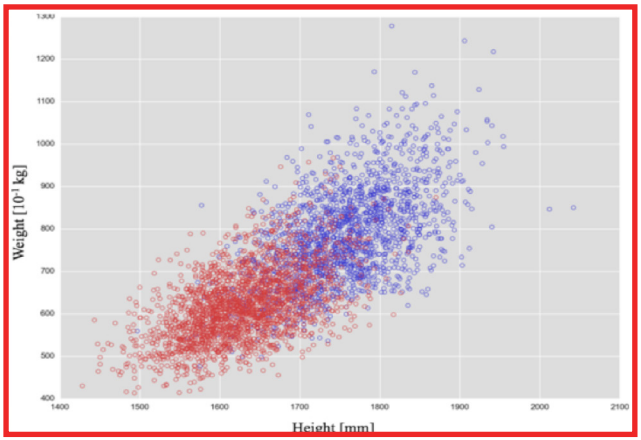
In our case, we are going to use the data found in a classic study that was conducted by the U.S. Army in 1988.

This study, called the 1988 Alpha Matrix Survey of Army Personnel, measured over 4,000 soldiers and then reported the data by gender. The data included over 100 measurements such as height, weight, length of fingers and so on. The Army's intent was to ensure that they could design equipment including helmets, trigger guards, or even tank seats, that would fit the soldiers.

We are going to use that data to train our machine and figure out if it can predict the gender of a soldier based on their measurements.

EXTRACTING THE RIGHT FEATURES

Our first question involves which measurements (or features, in ML terminology) can we pick to determine the gender? This is a very important step. Extracting the right features is key to the efficacy of our ML model. We'll start with something simple and intuitive, such as the weight and height of the soldiers and map those measurements on a graph.

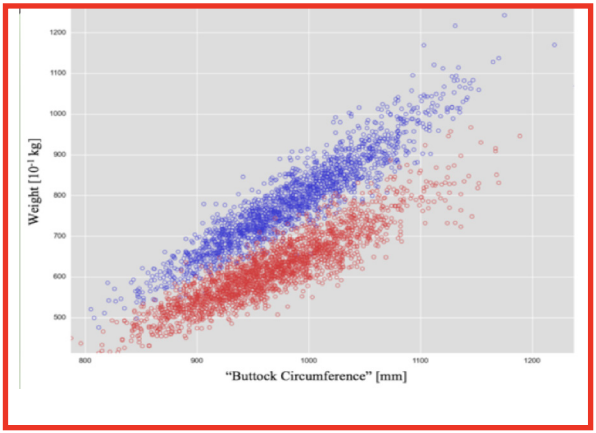


We can clearly see a blue and a red group, but there is a significant overlap in the middle, denoting an area that contains a combination of male and female soldiers. If we did not have data to tell us which is male and which is female, it would be difficult to correctly predict the gender of a circle located in the overlapping area. It means that height and weight are not a reliable feature pair to predict gender.

EXTRACTING THE RIGHT FEATURES

Buttocks Circumference	Weight

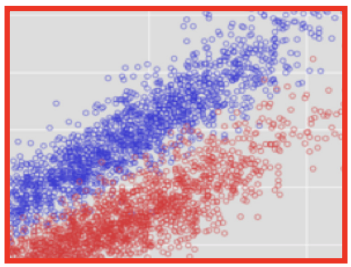
Since the Army survey contains over a 100 measurements, we can choose different ones. The data included a measurement called "buttocks circumference" in the survey. Instead of height versus weight, let's plot this feature against weight.



We see that the overlap is drastically reduced. We can conclude that the circumference of a soldier's buttocks is a more reliable feature than height.

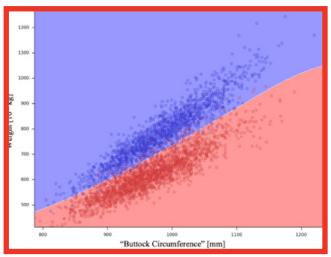
This demonstrates that in ML, the selection of features is critical, because certain measurements are better than others for separating elements.

CLASSIFYING — Running a Classifier



In this graph, we can easily distinguish the men from the women because we know the label: the red circles are female and the blue circles are male. But if all the circles were black, we'd have to use their location on the chart to guess their gender. To achieve that, we need to divide the chart into a male and a female area.

In ML, that step is known as running a machine learning classifier or classification algorithm. There are many types of ML algorithms, such as Bayesian networks, k-nearest neighbor, decision trees and so on. Each has its own advantages and might be better suited for specific tasks.

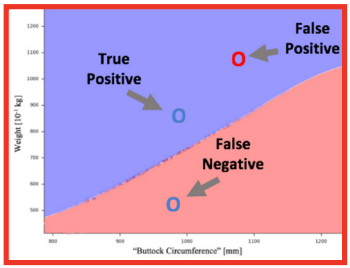


For our example we are going to run the algorithm, known as "support vector machine," against our data.

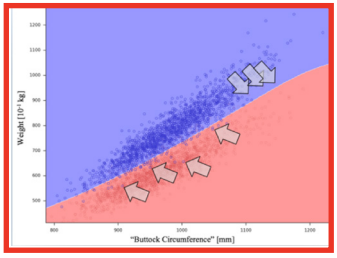
After running this classifier, we can clearly see the decision boundary in the middle. The algorithm will predict female for any circle in the red-shaded area and it will predict male for circles landing in the blue-shaded area.

CLASSIFYING — Balancing True Positives and False Positives

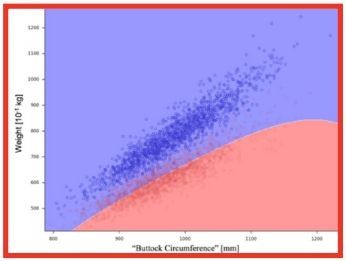
To address the issue of red circles in the blue area and blue in the red, we'll introduce the concepts of false positives and false negatives. Let's assume that we want to detect the male soldiers. In that case, blue is defined as our positive class.



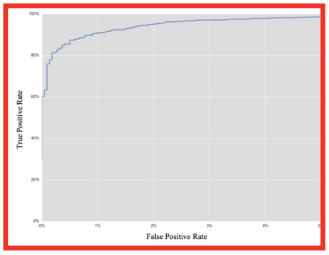
When blue is correctly predicted as male, we have a true positive. If a blue circle (known male) lands in the red area, it will be missed, since it has been classified as female. That's a false negative. Conversely, if a red circle falls in the blue area, it will be wrongly classified as male, which is a false positive.



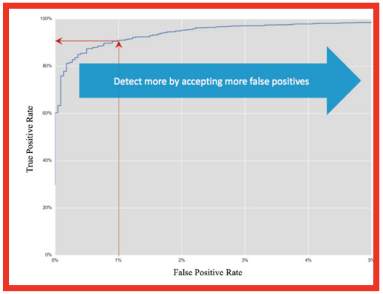
If we look at this graph closely, we can see some misclassification. This means that our model doesn't work in all circumstances. But we can improve our detection of male soldiers by pushing the decision boundary. The algorithm we chose allows us to push that boundary in either direction, so we are going to push it further and further into the red area. As a result, more and more blue circles will fall into the blue area.



This results in more blue circles in the blue area, or more true positives. However, we also get a lot more of the red circles showing up in the blue area. This means that we are getting more false positives. There is an inherent relationship between true positives and false positives: the number of false positives rises as the number of true positives increases.



There is a way to visualize that relationship in ML. It is called a receiver operating characteristic, or ROC curve. This curve plots the false positive rate against the true positive rate, showing the false positive rate against the prediction effectiveness.

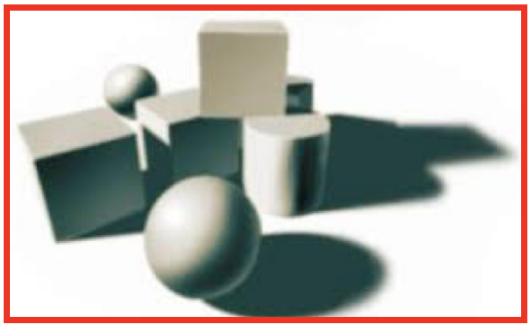


In this graph, we can see that tolerating one percent of false positives will yield 90 percent correct detections, or true positives. If we accept a higher false positive rate, the rate of true positives increases. This means that we can always detect more by tolerating more false positives.

CLASSIFYING — Adding More Dimensions

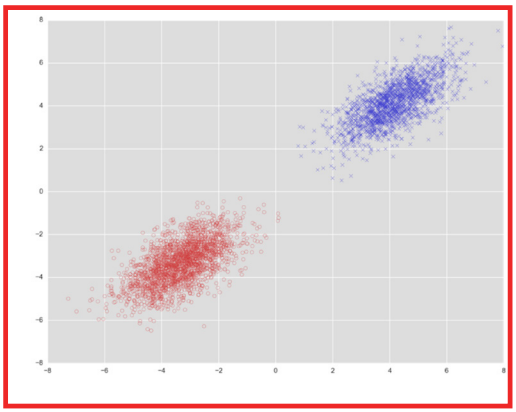
Height	Weight	Buttocks Circumference
Finger Length	Foot Size	Arm Length
...	...	...

We now have a working model, but it can be improved by incorporating more measurements, or features. So far, we have only used two of the hundreds of features that were recorded by the Army. In reality, ML can use thousands of features, also called dimensions.



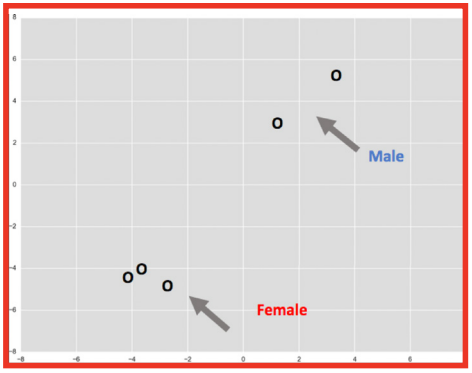
Since we can only visualize two dimensions on paper, we need to use a technique similar to casting a shadow in order to project multiple dimensions.

In the physical world, we can project the shadow of a three-dimensional object on a two-dimensional floor. We can do the same in ML, but instead of projecting three dimensions into two, we can project thousands of dimensions into two. And just like the shape of a shadow changes depending on the projection angle, the graph data will appear different, depending on how we project the measurements.



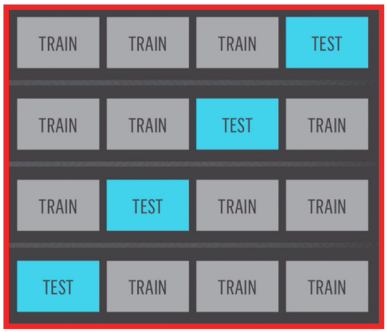
In the case of the army data, the model involves a high-dimensional space (over a hundred measurements). We are going to cast it on a two-dimensional plane, and just as with a shadow, if we cast it from the correct angle, we get a specific image. In our case, if we project from the correct angle, we can obtain a more accurate separation between male and female.

In this representation, we see all the female soldiers clustered in the bottom left and all the male soldiers clustered in the top right, and there is no overlap.



Now, if we obtain the measurements for an individual soldier, we can easily predict their gender, based on which cluster their measurements fall into.

CLASSIFYING — Adding More Dimensions



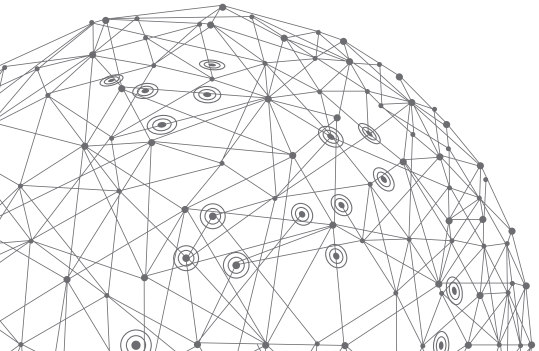
Once we believe we have a good working model, we need to make sure that it works as expected. For that purpose, we put aside a section of the original dataset for testing purposes. This data is not used for training, so the model can be tested with data it has never seen before.

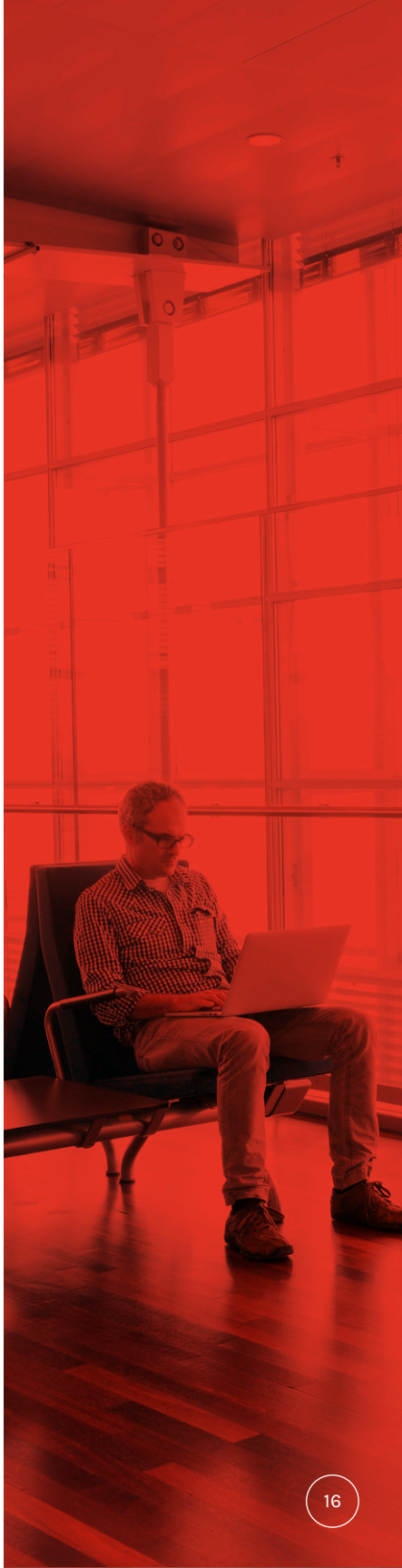
This allows us to measure the accuracy of the predictions. In order to be thorough with the testing, the model should be trained and tested multiple times, each time using different samples of the dataset. These steps allow us to fine-tune the parameters we use to train the final model of the classifier.

CHAPTER 3

MACHINE LEARNING APPLIED TO SECURITY: MALWARE DETECTION

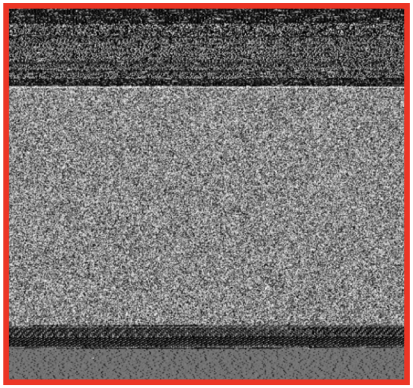
Now that we have seen how supervised machine learning works, let's see how CrowdStrike uses it as part of its malware prevention arsenal.





FINDING ENOUGH OF THE RIGHT DATA

In this example, we'll start by getting massive numbers of malware files from multiple sources, including industry feeds and CrowdStrike Falcon Intelligence™. Then we'll look at what's actually in each malicious file. We can use a grayscale to visualize the byte values in the file.

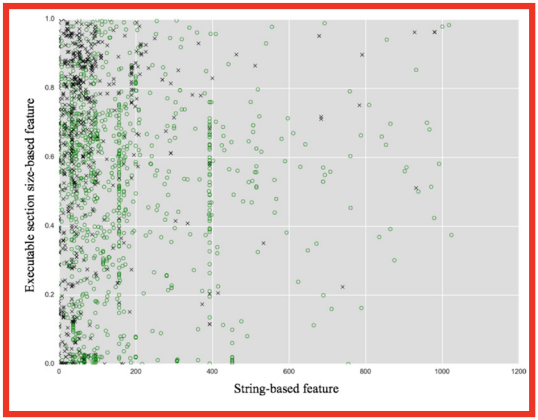


This is the visualization of a ransomware file. We can see some structure in the file: various blocks, or sections that appear different. We will capture these differences and use these measurements as part of the ML process.

EXTRACTING THE RIGHT FEATURES

32/64 BIT EXECUTABLE	GUI SUBSYSTEM	COMMAND LINE SUBSYSTEM	FILE SIZE	TIMESTAMP
DEBUG INFORMATION PRESENT	PACKER TYPE	FILE ENTROPY	NUMBER OF SECTIONS	NUMBER WRITABLE
NUMBER READABLE	NUMBER EXECUTABLE	DISTRIBUTION OF SECTION ENTROPY	IMPORTED DLL NAMES	IMPORTED FUNCTION NAMES
COMPILER ARTIFACTS	LINKER ARTIFACTS	RESOURCE DATA	EMBEDDED PROTOCOL STRINGS	EMBEDDED IPS/DOMAINS
EMBEDDED PATHS	EMBEDDED PRODUCT META DATA	DIGITAL SIGNATURE	ICON CONTENT	...

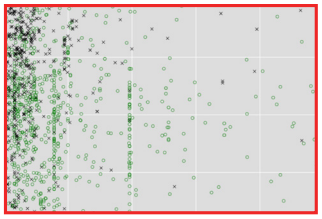
We can extract many features to capture the structure of the file at a very high level. Some features are obvious, such as file size and file entropy, or the amount of randomness in the file.



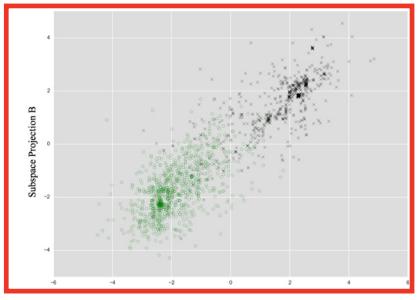
Let's take a look at two specific features: On the x-axis, we have a feature that we derived from printable strings extracted from the binary. On the y-axis, we have a feature that captures the size of a certain section of executable code in the binary.

Legitimate applications are plotted as green circles, and malware is plotted as a black X. The result is clearly much more disorganized than the Army's gender data.

CLASSIFYING — Adding More Dimensions



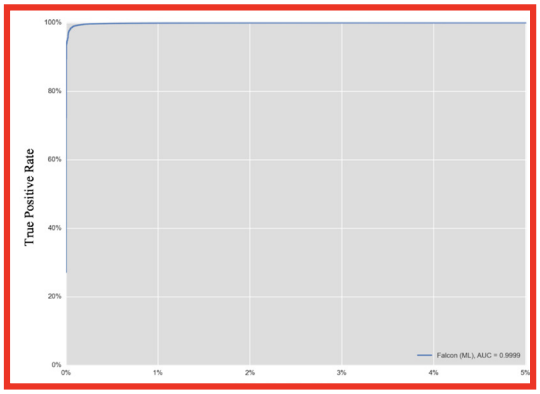
Although we can see some areas with a lot of green and relatively little black, the results are still very "noisy." This means that the data is a lot more complicated than the previous example, but just as we did for the army data, we can add more features to help.



So, let's select and combine a couple of hundred features and project those 100-dimensional features on a two-dimensional plane. Now we can see some clusters emerge. All the legitimate applications are now on the bottom left and the malware is on the top right. However, there is still some overlap. We did not obtain a clean separation like we had with the army data.

Unfortunately, there is no angle that would give us complete separation when we cast a shadow. This is because there is no clear delineation between malware and a clean application. To solve this, we need to project the measurements onto a curved surface, rather than a flat plane (to understand the difference between the views, think of how a flat map of the world is different from a globe). Since we are working with high-dimensional space, we will be able to find a curved surface that separates these two areas allowing us to have malware appear on one side of the surface and the clean files on the other side.

CLASSIFYING — Balancing True Positives and False Positives



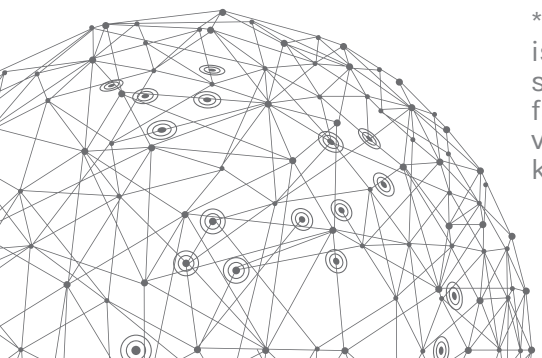
The ROC curve in the graph on page 12 shows the efficiency of the model using just two features of army data.

In this graph, we see the ROC curve for the CrowdStrike ML engine. It is exactly the type of ROC curve that is desired. The barely visible blue line shoots straight up, showing it detects the whole sample of malware files with minimal false positives, and then at the very top, it slants out further to the right. One of the key efficiencies is a metric called "area under the curve." In our example, the area under the curve covers 99.9 percent of the graph, almost its entirety. This means that our ML engine detects malware with extremely low rates of false positives.

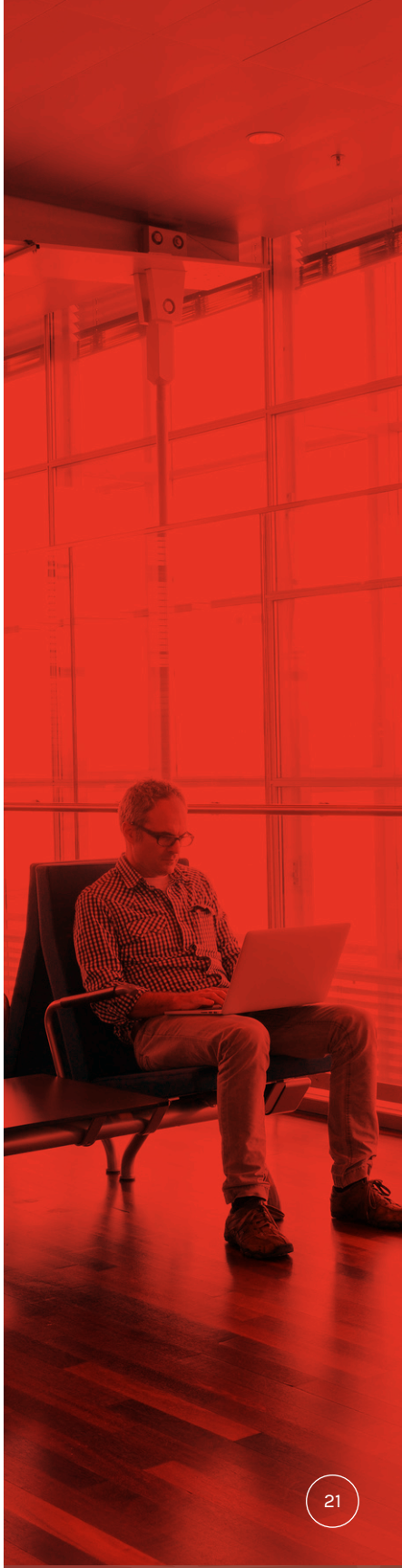
CHAPTER 4

REAL WORLD EXAMPLE: CROWDSTRIKE MACHINE LEARNING SCANNER IN VIRUSTOTAL

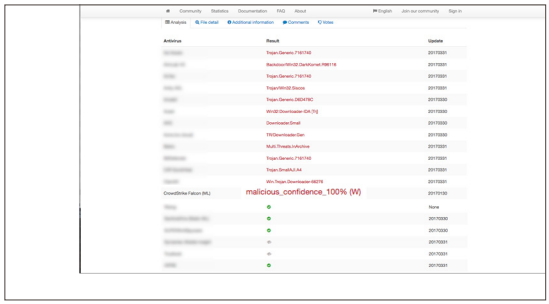
Everyone — even those who aren't CrowdStrike customers — can enjoy the benefits of CrowdStrike's ML-based malware detection, as it was recently adopted as the first purely ML-based scanner technology to be incorporated into VirusTotal*.



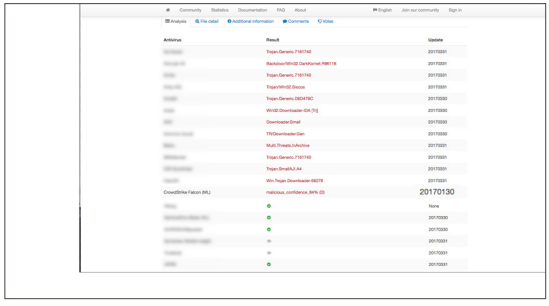
*VirusTotal, a subsidiary of Google, is a free service that analyzes suspicious files and URLs to facilitate the quick detection of viruses, worms, trojans, and all kinds of malware.



This means that users can go to virustotal.com, upload a file and have it scanned by CrowdStrike's ML file analysis — at no charge. If you try this, you'll notice a couple of differences between the CrowdStrike ML scanner and other scanners. First, CrowdStrike returns a confidence score.



Instead of a simple yes or no answer as to whether the file is malicious, the CrowdStrike engine gives users an indication of how certain it is that a given file is malware (see page 27 for details).



A second difference is that since the ML engine is signatureless, it doesn't need to be updated. If we look at this screenshot, we can see that CrowdStrike's ML engine is 77 days older than the most recent of the other malware detection engines listed, all of which rely on signatures.

This is because unlike traditional AV engines, CrowdStrike's ML-based engine still works even though it isn't updated on a daily basis. In ML terminology, we say that the model "generalizes." It means that instead of having to memorize a set of specific malware file signatures, ML can learn without being fed a new data set every day. As a result, it can look at the broader picture — the high-level traits — to decide if a file is malicious.

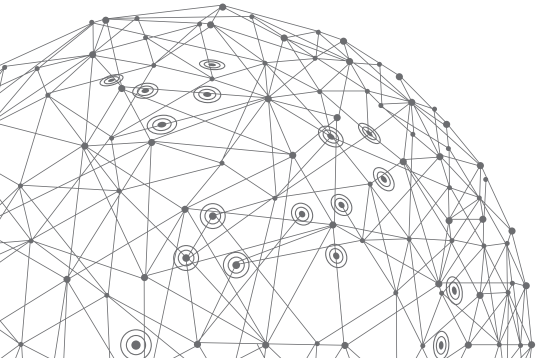
The ML scanner is just one aspect of the protection offered by CrowdStrike Falcon®, which incorporates a mixture of potent technologies for endpoint protection. This is important to understand because, in spite of its advantages, ML is not perfect.

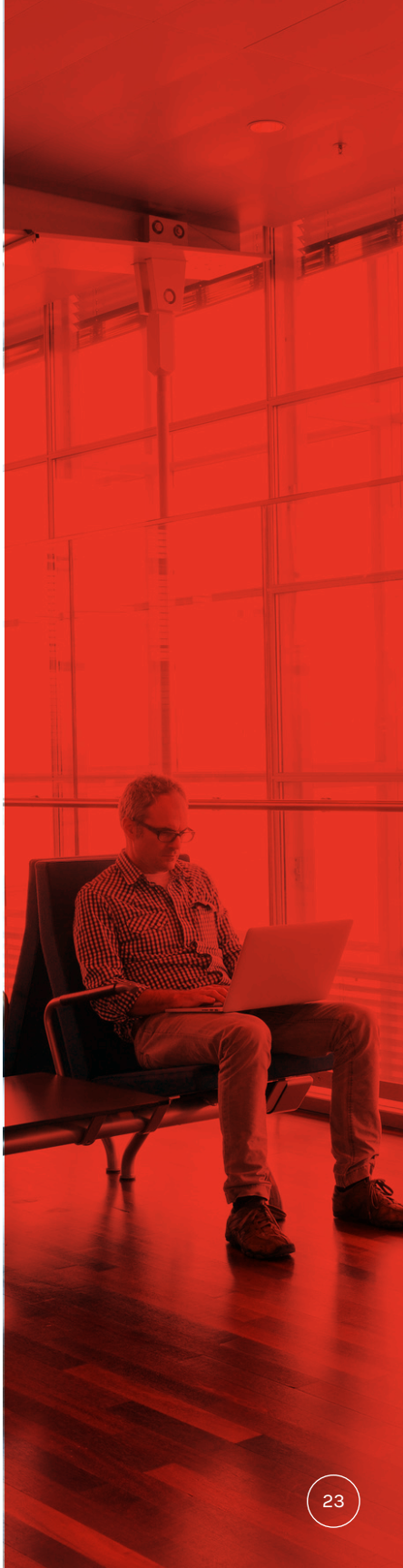
CrowdStrike's machine learning scanner is just one aspect of the comprehensive protection offered in the Falcon platform

CHAPTER 5

THE LIMITATIONS OF MACHINE LEARNING

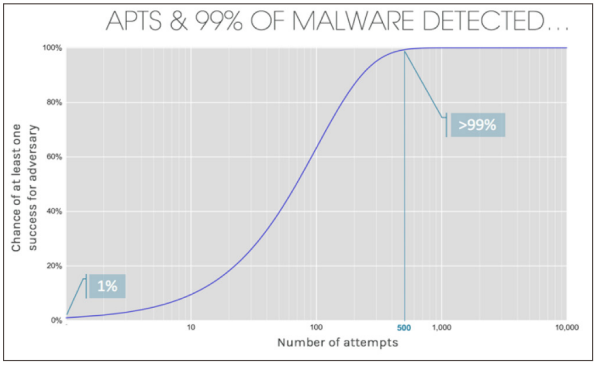
Machine learning can increase your ability to stop sophisticated threats, but it does not solve all security problems.





To understand the limitations of ML, let's take a look at what malware detection rates mean, in terms of protection against advanced persistent threats (APTs). Let's assume that we have an ML engine that detects 99 percent of malware. In fact, according to third-party testing organizations (such as AVtest.org), the industry average detection rate for prevalent malware files is 99 percent, and the detection rate for zero-day, or unknown malware, is 98 percent. So, let's use 99 percent detection as our base number.

To succeed, an adversary conducting an advanced persistent attack must get at least one of their malware files to run in the victim's environment. Once they do, they will have established a beachhead that allows them to move laterally in the victim's network. If the attacker sends only one malware file to the victim, they have a one percent chance of success, since 99 percent will be detected. If we visualize the attacker's probability of having that one file bypass the 99 percent efficacy of the ML engine, we obtain the graph below.



As this curve shows, the attacker's chances grow rapidly as more malware files are directed against the target. At around 500 files, the attack has more than a 99 percent chance of succeeding. This means that all an attacker needs is 500 unique malware files to achieve a more than 99 percent chance that at least one of his files will make it into the victim's environment.

All an attacker needs is 500 unique malware files to achieve 99+ percent certainty that at least one of those files will make it into the victim's

That volume of malware files is not hard to achieve for most APT actors, including many that have the resources of a nation-state for support. However, for this volume-based approach to work, the malware files must be written without sharing any information, so that each is entirely unique. In such a case, there is absolute certainty that one of these malware files will succeed in penetrating the ML engine's defenses. We have made some simplifications here to prove the point, but it's clear that operating on its own, even a highly effective ML engine can be defeated.

Unfortunately, the problem is compounded by the fact that only 40 percent of breaches are malware-based and the majority, 60 percent, use non-malware-based techniques. That 60 percent includes intrusions that use exploitation techniques, those that use stolen credentials, and attacks from sophisticated actors such as nation states or organized criminal gangs. We just saw how the 40 percent of malware-based attacks are nearly impossible to block if you have an adversary with resources, but you also need to worry about the additional 60 percent of intrusions that don't use malware.

60 percent of intrusions don't use malware.

ML's Best Fit Is to Be Part of a Comprehensive Solution

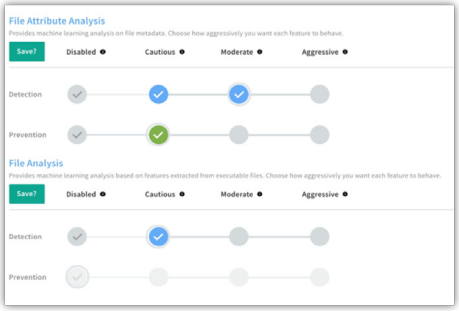
To address that challenge, an ML analysis engine needs to be part of a broader solution. This is why CrowdStrike's next-generation antivirus component of the Falcon platform includes multiple complementary techniques in addition to ML. These techniques include exploit prevention, and behavioral analysis. CrowdStrike's unique approach is based on indicators of attack (IOAs). IOAs are determined by analyzing the behavior of events and actions to detect the attacker's intent, regardless of the malware or exploit used in an attack.

Although IOA analysis is highly effective, especially when combined with ML-based malware detection, an extremely motivated attacker with plenty of time and resources could still eventually succeed in penetrating a victim's network. A comprehensive endpoint protection platform also needs to provide effective defense even when an intrusion occurs. That's why Falcon includes advanced endpoint detection and response (EDR) capabilities. EDR provides the visibility you need to see what's happening on your network, what the adversaries are doing, and to stop them before they can cause serious damage.

Conversely, EDR can be used to hunt through endpoint data to see if anything out of the ordinary is happening on your network. This kind of proactive hunting is vital to stopping highly sophisticated attacks orchestrated by advanced adversaries. To that end, CrowdStrike offers a managed hunting option that matches a team of dedicated security experts against sophisticated adversaries. The CrowdStrike managed hunting team, called Falcon OverWatch™, sifts through large volumes of EDR data to find novel attacks and uncover entrenched actors whose stealthy activities can otherwise go undetected for a long period of time.

PUTTING MACHINE LEARNING IN PERSPECTIVE

Within the Falcon platform, the ML techniques are exposed in the form of a slider.



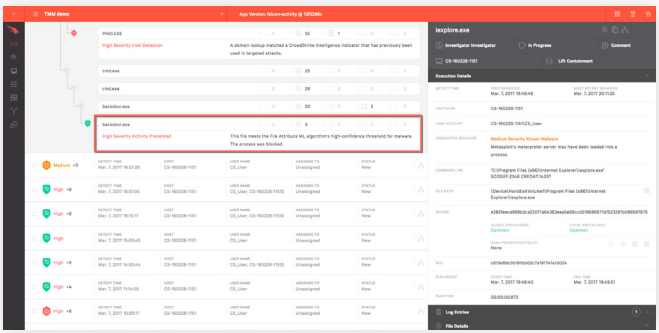
Customers can fine-tune the level of aggressiveness for these ML algorithms. They can choose to be cautious, which means files will be "convicted" only if the algorithm is highly confident, or they can be

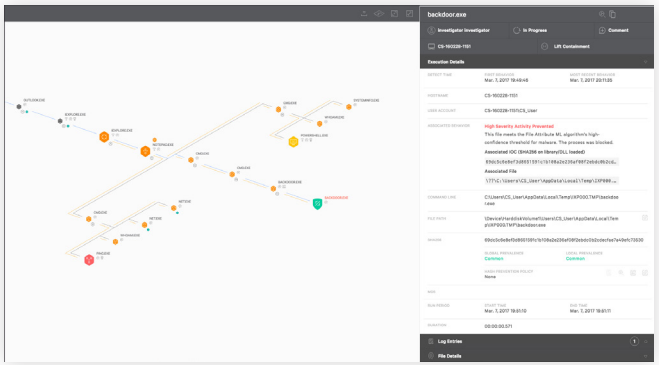
Machine learning's best fit is to be part of a comprehensive solution.

more aggressive, which results in convicting more files while increasing the risk of experiencing a few more false positives. Users can also choose separate sensitivities for detection and prevention. This allows customers to be more cautious when blocking malware from executing, for example, while retaining more aggressive alerting.

To tie this back to ML concepts, CrowdStrike Falcon allows customers to balance the incidence of true positives versus false positives by pushing the decision boundary with the slider.

The screenshot below shows what an ML alert looks like in the Falcon user interface. We can see in this example that CrowdStrike's ML engine detected and prevented the execution of malware. In addition, if a customer wants more data around that incident, they can search and browse the event database and process explorer, two EDR capabilities which allow a security team to "go back in time" and view all the context and events that led to the attempted execution of this malware.





CONCLUSION

Machine learning is an effective tool against both known and unknown malware because when applied correctly, it can understand and identify maliciousness. ML doesn't have to memorize signatures, it understands and applies the concept.

However, not all machine learning is created equal. In order to perform effectively, or to achieve an acceptable balance between true and false positives, an ML engine needs to get the right data, extract the right features and cast the right angle on those features. In summary, if the machine is trained poorly, it will produce wrong predictions.

Finally, machine learning is an important weapon in the endpoint protection arsenal, but it doesn't solve all problems. Attackers can use brute force to bypass even the best ML engine, by sending large quantities of unknown malware or by using techniques that don't use malware.

As a result, endpoint solutions that rely solely on ML techniques will ultimately fail. CrowdStrike Falcon, offers peak effectiveness because it combines ML and other next-gen AV technology with robust EDR and managed hunting capabilities. The result is a comprehensive solution that protects against the full array of threats facing today's organizations: from "commodity" malware to advanced zero-day threats, and even the most sophisticated malware-free attacks.

ABOUT US

CrowdStrike® is the leader in cloud-delivered next-generation endpoint protection. CrowdStrike has revolutionized endpoint protection by being the first and only company to unify next-generation antivirus (AV), endpoint detection and response (EDR) with Falcon Insight™, and a 24/7 managed hunting service, Falcon OverWatch™, all delivered via a single lightweight agent. The CrowdStrike Falcon® platform, certified to replace legacy antivirus, has reinvented how endpoint security is delivered with its industry-leading, cloud-native architecture. The CrowdStrike Falcon platform protects customers against all cyber attacks, using sophisticated signatureless artificial intelligence/machine learning and indicator-of-attack (IOA) based threat prevention to stop known and unknown threats in real-time. Core to its innovative approach is the CrowdStrike Threat Graph™, which analyzes and correlates over 34 billion events per day from millions of sensors deployed across 176 countries, uniquely providing crowdsourced protection for the entire customer community.

Many of the world's largest organizations put their trust in CrowdStrike, including three of the 10 largest global companies by revenue, five of the 10 largest financial institutions, three of the top 10 healthcare providers, and three of the top 10 energy companies.

CROWDSTRIKE

crowdstrike.com

15440 Laguna Canyon Road, Suite 250, Irvine, CA 92618

VER. 03.27.17