Advanced Analytics for Technology Risk

Também disponível em português
www.isaca.org/currentissue

Over the decades, information systems security has evolved in many ways. The problems have changed and the tools have, too. While in the past, the focus was on audit and internal controls that were more inward looking, the risk that IT risk professionals find themselves protecting against and spending the most hours on is cybersecurity risk. Because of the digitization of everything, a processor in everything and, therefore, the presence everywhere of code that can be compromised, the connected world today presents threats that are very different from those faced in the 1980s and 1990s. The tools have evolved, too, from basic computer-assisted audit techniques (CAATs) to more advanced analytical approaches.

An Overview of Advanced Analytical Tools

For a long time, starting in the 1950s and 1960s, many experts believed that human-level intelligence could be achieved by having a large enough set of explicit rules that manipulate knowledge. This approach was known as symbolic artificial intelligence (AI) and was the dominant way to think about AI for a long time. Though this approach was useful for well-defined problems that could be expressed as complex sets of if/else rules, it was difficult to apply to less predictable tasks such as language translation or image classification. Even for problems where this approach was attempted, it turned out that the rules proved expensive to maintain and the entire approach was rather brittle.[1] Machine learning steps away from defining precise rules, instead choosing to convert the problem into data and then apply statistical methods to these data to obtain needed results.

Machine Learning, AI and Natural Language Processing

Machine learning is an effort to automate what are considered intellectual tasks that are normally performed by humans. It allows computers the ability to perform tasks without being explicitly programmed.

When it is said that machine learning is about discovering relationships between data, often this relationship is expressed in a way that a set of variables can be used to determine the value of another variable (that is unknown and of interest to know). For example, if characteristics of an email message (i.e., the count of $ signs, the extent of the use of uppercase letters or the presence of links) are known, can it be determined whether the email is a phishing attempt? Similarly, if the attributes of an application (i.e., the results of the last business continuity plan tests or the characteristics of upstream dependencies) are known, can the expected downtime in the event of an outage be estimated?

The first question is the example of a classification problem and the second of a prediction (or estimation) problem. The difference between classification and prediction is that, in the former, a bucket or category to which an item belongs is

Do you have something to say about this article?

Visit the *Journal* pages of the ISACA® website *(www.isaca.org/journal)*, find the article and click on the Comments link to share your thoughts.

https://bit.ly/2Us3jEo

Mukul Pareek, CISA, ACA, ACMA, PRM
Is a risk management professional based in New York, USA. He is copublisher of the *Index of Cybersecurity* (*www.cybersecurityindex.org*) and the author of the risk education website *www.riskprep.com*. He has been published on multiple topics relating to risk measurement in the *ISACA® Journal*.

identified (e.g., phishing or not), while in the latter, a number is predicted (e.g., downtime in minutes). For the machine to be able to learn such classification by discovering the patterns that allow it to estimate downtime or email type, it needs to be provided with a sample set of data where this information is already known. This labeled data set is called the training data set. Using these data, the machine identifies the patterns that may not be observable to the human eye because of the number and variety of variables. This type of learning is called supervised learning as the computer learns from an already classified and labeled data set. When provided with a new, unlabeled entry based on the patterns the computer model sees in the pre-labeled data it learned from, a prediction can be made for the new observation for which the estimate or classification is not known.

If this sounds similar to old-fashion regression modeling, it actually is, because variants of regression analysis underlie many different machine learning algorithms. In fact, one could argue that an overly simplistic explanation of a neural network would be to describe it as a series of linear regressions stacked sequentially.

In addition to prediction and classification, there is yet another type of problems called clustering problems that machine learning can solve. As an example, if someone has a list of Internet servers with their IP addresses, the number of websites they host, the operating systems they run, the web server software, whether they host a mail server, where they are located, the number and types of top-level-domains they each run, etc., then how can these be bucketed together so that the servers that are the most similar to one another are in a single bucket? In this situation, machine learning algorithms create buckets, called clusters, to which each server would belong. The algorithms do so without any input by the human, hence, this type of learning is called unsupervised learning.[2] These clusters can provide a human analyst insight as to the kinds of servers, possibly with a view to support a decision to block some of them. Clustering is an example of unsupervised learning because it does not require an existing example set from which to learn.

In reality, when one peeks behind the curtain, the phrase "machine learning" is a bit of a misnomer. In all of it, there is no machine that is learning anything in the sense that people comprehend learning. Most of it is mathematics applied to data to discover relationships within a data set. These relationships are then used to create applications that perform complex cognitive activities, such as image recognition or speech generation. With the state of the science today, the prospect of machines teaching themselves general intelligence and becoming sentient and taking over the world is highly unlikely.

These developments have significant implications for risk management, particularly IT risk. Questions relating to model governance, change control over models, access to data sets and resilience in situations where a model is unable to deal with unpredictable exogenous changes need to be considered by IT risk professionals. At the same time, risk professionals can adopt these technologies to scale and automate their own work.

> "MANY POPULAR ALGORITHMS WIDELY EMPLOYED TODAY WERE CREATED DECADES AGO (INCLUDING NEURAL NETS), AND SOME ARE MORE THAN HALF A CENTURY OLD."

While machine learning has recently undergone widespread adoption across industries and fields of study, the subject is not entirely new. Many popular algorithms widely employed today were created decades ago (including neural nets), and some are more than half a century old. A great deal of confusion also results from multiple phrases used to refer to similar things, e.g., data mining, statistical learning, machine learning, AI, deep learning, shallow learning and narrow AI.

Of course, for the pedantically focused, it might be possible to identify fine differences between all these, but the reality for the practitioner is that in a practical sense, these are interchangeable terms when it comes to creating applications. The problems solved, i.e., prediction, classification and clustering, are roughly the same. If one uses plain vanilla linear regression, it might be called data mining, but if a neural network is used, it might be called deep learning.

However, given the press about AI, it might be worthwhile to discuss that in a bit more detail.

AI is a general field that encompasses machine learning, of which deep learning is a part. Computer programs that use a representation of known knowledge as hard-coded rules, no matter how complex, would not be called machine learning. Therefore, earlier chess programs would not qualify as machine learning though they would fall within the larger concept of AI.

A subset of machine learning is deep learning, specifically models that use neural nets. The term “deep learning” does not indicate any kind of deeper understanding of information than in general intelligence, but that multiple layers of data, with each layer containing a transformation that feeds the next layer, are used. The use of multiple layers allows neural nets tremendous flexibility as these layers can essentially represent any function. Neural nets have been used to achieve human-level image and speech recognition, the ability to answer natural language questions, and above-human-level go playing. While these seem like magical accomplishments, these are still far from what is considered general intelligence of a kind humans display every day. Because neural nets underlie many of these advanced uses, neural nets, or deep learning, are occasionally equated with AI. However, what these techniques are called is immaterial as long as how to apply and implement them is known. As a result, the terms “data science,” “machine learning” and “advanced analytics” will be used interchangeably.

Yet another field of interest that is in the mix is natural language processing (NLP). NLP is concerned with deriving meaning from text and expressing meaning in the form of text. This processing involves translating natural language into data that can then be processed to derive meaning, identify topics, determine sentiment or generate text in response. NLP applications are found everywhere and include spell checks (rule based), chatbots, spam filters, text summarization, plagiarism detection, sentiment analysis and creative writing.[3] Most NLP applications, though not all, use neural networks. Alexa, Amazon’s voice assistant, uses a service called Amazon Lex, which needs an extremely broad capability to respond. The logic, however, is shallow: a set of trigger phrases that all produce the same response based on a single-layer flat if/else tree. The NLP field is continuing to advance rapidly.

Model Building

As discussed, machine learning is about discovering patterns in data. These patterns are expressed as a variable of interest being predictable using other variables or attributes, often called features. Someone may, for example, be interested in predicting whether a given network activity is malicious or not, and the features that may be used to make this prediction may include IP address, types of requests, packet sizes, etc.

> “THE PROCESS WHERE THE RELATIONSHIP IS ESTABLISHED IS CALLED THE LEARNING. ONCE THE RELATIONSHIP IS KNOWN, IT IS EXPRESSED AS A MODEL.”

Going back to the regression analogy, it is not very different from a variable *Y* described as a function of a number of features (i.e., $Y= f$ [feature 1, feature 2,…feature *n*]) where *Y* would be whatever someone may be trying to predict, and the various feature variables are the independent variables. However, this relationship is encapsulated in a model and is generally incapable of being expressed as a closed-form mathematical function. A model is the artifact created by the machine learning process. Once the model is built, a set of features can be passed to it, and it will provide a prediction.

Machine learning starts with data. Think of data as being a large table in a spreadsheet. There are columns, and there are rows. As an example, consider a model that needs to classify mail server or web server connections as being valid attempts or malicious ones. On the mail server side, the relevant features to be extracted, or columns in the spreadsheet, might include user login time and date, IP address, geographic location, email client, administrative privileges, and Simple Mail Transfer Protocol (SMTP) server activity. On the web server side, the relevant features might include user IP address and location, browser version, the path of the pages being accessed, the web server status codes, and the associated bandwidth utilization.[4] To train the machine learning model, the data need to be pre-classified as being valid or malicious attempts. These data are called labeled data, i.e., for each row, the answer is already known. This allows the data to be used as training data, i.e., they will enable the algorithm to determine the relationship between the known and unknown variables. The process where the relationship is established is called the learning. Once the relationship is known, it is expressed as a model.

The model can then be used to predict whether future logins are malicious or otherwise by passing to it all known features for these logins.

For machine learning to work, a few things need to be in place:

- **Training data**—Training data are the input data with the correct or expected outputs already provided. Data may come not only in their familiar form as database or spreadsheet tables, but also as images, sound files, log data, etc. A subset of attributes that are relevant to the prediction is identified. The training data need to contain the labels for the output of interest. For example, if classification is being solved between malicious and nonmalicious access attempts, historical data that are already labeled would be needed to "train" the machine learning model.
- **Model building**—This is the sequence of mathematical steps that is used to identify the predictive relationship and express it as an artifact called the model. There are hundreds of machine learning algorithms available to choose from: logistic regression, random forest, gradient boosting models, support vector machines and so on. When people speak of neural nets, which are also algorithms, there are various types of neural nets such as convolutional neural networks for image recognition, recurrent neural nets for natural language processing, etc. The algorithm to use is decided based on the use case. Often, multiple algorithms are tried, and the one providing the best fit is selected.
- **A metric to measure model effectiveness**—For classification problems, often the accuracy rate or the proportion of the sample correctly classified is the metric to measure the effectiveness of the machine learning model. For estimation problems, where a number needs to be predicted, people think of how far the predictions are from actually observed quantities (technically measured as root-mean-square-error [RMSE]).

It is not necessary that only one model is used to build a solution. It is possible to combine multiple models together to achieve superior results. This approach, called ensemble methods, leverages multiple algorithms simultaneously. For example, many algorithms are used to make a prediction, and then the value predicted by the majority of the algorithms in the ensemble is used as a final prediction.

The Democratization of Data Science

Over the past few years, data science has been democratized by the availability of open-source software, tools, libraries and educational resources. A whole generation of citizen data scientists have trained themselves on these tools and are contributing to the development of the field across the globe. Compute-intensive hardware is no longer out of reach of the average organization. Graphical processing units (GPU), which are capable of performing fast calculations, are now available for less than US $1,000. The ability to spin up a fast analytics machine on Amazon Web Services (AWS) or the Google Cloud Platform in a matter of minutes for less than a dollar an hour is available to anyone with an Internet connection.

Model building, and predictive analytics at large, are fast approaching commoditization. The means to run data through an algorithm and tune it to perfection can be used as commodities. The skills to do all of this are fast becoming commodities as well. In the near future, it will not be a surprise if learning algorithms are packaged as point-and-click options in desktop spreadsheets.

The implications of this are significant. First, technologies that were once available only to the largest of organizations are now available to every individual and any organization. Second, mid- to senior-level managers have a responsibility to understand the capability of these technologies and, specifically, be able to identify use cases where these can provide an advantage. Organizations that can build a strong data culture will be able to move forward much faster.

> MODEL BUILDING, AND PREDICTIVE ANALYTICS AT LARGE, ARE FAST APPROACHING COMMODITIZATION.

Other than in large technology enterprises such as Google or Facebook or other unique situations, most data scientists at organizations are not creating new algorithms. The algorithms mostly already exist. Most practical benefits can be gained by using what is already available. That is not to say that new inventions are not happening: Google, IBM and others are doing that, but, by and large, the models already exist. The problem is generally reduced to determining which model works the best and gives the most accurate results. These models exist inside of software libraries and packages, most available as open-source software. As an example, the caret package[5] in R (which is an open-source statistical software) provides access to more than 200 machine learning models. Similar things can be done in Python, another open-source programming language.

Of course, the machine will not pick the model; data scientists use their knowledge of the data to decide which will work best in the given situation. Even this decision-making is being automated—there are routines available that will run a data set through all possible models and show which one gives the most accurate results.

The Challenges

A number of common problems vex most data science projects across organizations.

The first challenge is to get good-quality labeled data to use for training, i.e., for the model to learn. Labeled data means existing data that are already classified so the machine can make a good estimate of the relationship it needs to predict. It is here that large enterprises such as Google and Facebook have a major advantage compared to other organizations, as they have unparalleled access to high-quality data. As an example, each time a user overrides Google's autocorrect on a phone, Google gets more labeled data to figure out what is correct. Most organizations cannot compete with that.

The second is an engineering challenge. Data rarely come neatly organized as a table in a database or as a clean spreadsheet. They come as feeds from sensors, log files, emails, images, videos and conversations. A lot of heavy lifting needs to be performed to organize this unstructured data into formats to which algorithms can be applied. Data need to be taken through a lengthy process of preparation, cleaning, addressing missing or incorrect values, and they have to be transformed, joined and wrangled in multiple ways. Once the data are available, feature extraction needs to happen, which is the process of identifying the variables that can help predict whatever it is that needs to be predicted. The speed of all of this can be an issue for anything that strives to be near real time.

A third challenge is that models perform multiple computations in a way that it becomes difficult to explain their output. This creates a credibility challenge as users are being asked to trust a system without a logical step-by-step connection available between the inputs and the outputs. As users become familiar with the system and trust grows, this challenge is reduced. Data scientists often choose simpler but less accurate models for the sake of explainability. Model explainability is being addressed using concepts such as local interpretable model-agnostic explanations (LIME), which have been implemented in Python and R libraries.[6]

Use Cases Within IT Risk

IT risk professionals have always been users of analytic methods. The use of audit risk tables, sampling techniques and other approaches have roots in statistics, while the use of CAATs (notably using ACL) has been around for decades. However, the use of advanced analytical techniques opens up a completely new area for exploration for technology risk, the effective implementation of which requires imagination, education, creativity and audacity. This is because traditional analytics have, by definition, been backward looking, i.e., they provide a lens into the past. Historical data are sliced, diced and summarized in multiple ways and visualized using dashboards. Many of these traditional analytics,

which have been based on strong data extraction, transformation, and loading processes and engineering foundations, continue to provide valuable insights into risk posture, remediation, root causes, patterns and trends, and will continue to be the bedrock of supporting the risk manager.

However, these new analytical tools allow a completely different angle of view, which is the ability to probabilistically predict the future. Data science techniques can provide probabilistic estimates of what can go wrong, which, when combined with the intuition and judgement of the risk manager, can be a potent weapon to better manage risk.

> “THE USE OF ADVANCED ANALYTICAL TECHNIQUES OPENS UP A COMPLETELY NEW AREA FOR EXPLORATION FOR TECHNOLOGY RISK, THE EFFECTIVE IMPLEMENTATION OF WHICH REQUIRES IMAGINATION, EDUCATION, CREATIVITY AND AUDACITY.”

All risk management focuses, in some form or another, on avoiding adverse risk events. These events run the gamut from data leakage to server compromise, from failed IT changes to misused privileges and from intrusion detection to flagging spam.

A caveat to keep in mind is that data science cannot solve all problems. Having petabytes of data may be useful, but not all data will have predictive power. There is only one way to find out though, which is the scientific method of testing and verifying. Sometimes, exploring data science models helps uncover associations that were not known before, and these associations may provide a rule-based solution when a model-based approach may be considered overly complicated. An approach that may create insights for one organization may not provide anything useful to another. This is because the situation, the data and the context could all be different.

What Does Production Look Like?

Data science projects, when applied to technology risk, need to provide decision support. The practical manifestation of the work done as a part of a data science project may come in a variety of forms. The most common of these include:

- **Information in reports or dashboards**—This is often how most new data science projects get implemented. The results of the data science model or analysis are published as data on dashboards or other visualizations, where they can be advantageously combined with traditional analytics. Sometimes, they can be published as reports or alerts emailed to the consumers of the information. This is generally the fastest and most popular way to get to production. However, if these reports or dashboards are not part of the risk workflow or impact incentives in some way, they are likely to fall into disuse over time as the novelty wears off.
- **Embedded in other applications**—Requiring greater integration than the reporting or dashboards approach, the results can become a part of a traditional application, creating a tighter workflow integration of the team’s data science capabilities. For example, a model may, in real time, predict a risk rating for an issue, change request or incident as the transaction is being recorded in the system.
- **Automatic decision-making**—In these cases, the system may make a decision and implement it, directly performing the role of a human who, in the past, may have made such a judgement. Automated credit decisions based on features such as an applicant’s history incorporated into online credit application tools are an example of learning-based decisions. In the world of technology risk, such decision-making could be used to drive escalations of incidents or other transactions requiring human intervention.

IT Risk Use Cases

The following are some examples of possible use cases for machine learning and data science projects for managing IT risk:

- **Data leakage**—A key risk for any enterprise, data leakage detection tooling can be built using machine learning, leveraging email content and metadata, printer logs and access. The larger vendors offer data loss prevention (DLP) as a service, with many of the components built in (e.g., Amazon’s Macie and Google’s DLP application programming interface [API] for enterprise Gmail customers). These can accelerate improving DLP practices using the

> “THESE NEW ANALYTICAL TOOLS ALLOW A COMPLETELY DIFFERENT ANGLE OF VIEW, WHICH IS THE ABILITY TO PROBABILISTICALLY PREDICT THE FUTURE.”

pre-trained templates from these vendors for common types of sensitive information.

- **Service downtime**—Service downtime can be modeled using logs from the OS, middleware, database and applications. Log data can be diverse, voluminous and often comes in the form of time series data requiring special data prep techniques. If done well, models to predict downtime can be enlightening as to underlying associations.
- **Failed IT changes**—Systems failures are often the result of failed IT changes that have been scheduled and planned in advance. Modeling techniques can provide an advantage over rule-based assessments by predicting which change requests are likely to result in downtime, rollbacks or other adverse impacts.
- **Spam and phishing detection**—Had it not been for spam detection using NLP techniques, email would have been dead by now. Spam and phishing detection are primary use cases that are in operation for nearly everyone, whether they use a custom solution or a provider such as Gmail.
- **Intrusion detection**—Most intrusion detection systems leverage packet-level data (pcap files) across the various types of Internet protocols such as Transmission Control Protocol (TCP), User Datagram Protocol (UDP), Internet Control Message Protocol (ICMP) and others. Packets may encapsulate other higher-level protocols as their payload such as Hypertext Transmission Protocol (HTTP), Post Office Protocol (POP), and Network File System (NFS).[7] Netflow data that capture dataflows between given IP addresses are another valuable source of data for detecting network intrusions. Clustering, neural networks, ensemble methods and various other methods can then be employed to detect anomalies that may be investigated for malice.
- **Privilege misuse**—Privileged session monitoring that is often enabled on sensitive hosts can create a large amount of data for which the cost of a human review may be completely infeasible. An appropriately trained machine learning model can help identify departures from normal privileged activities for further investigation.
- **Third-party risk**—A recent study[8] looks at predicting the chances of a breach for any organization based on externally observable properties of the organization’s network. Using features such as misconfigured open DNS resolvers, untrusted certificates, open SMTP relays, inclusion in public malicious activities lists such as SpamCop, PhishTank and other sources, the authors were able to model and accurately predict future breaches with a nearly 90 percent true positive rate, a 10 percent false positive rate and an overall accuracy of 90 percent.
- **Hardware failures**—Hardware logs, incapable of human review and interpretation, are just the right feed for a machine to analyze patterns to predict failures in advance.
- **Network activity on mail and web servers**—Network activity on mail servers can be analyzed using, for example, user login time and date, IP address, geographic location, email client, or SMTP and POP activity. Network activity on the web server side can be analyzed for similar attributes and browser version web server status codes, pages accessed, bandwidth utilization, etc. Models can predict and proactively block sessions deemed malicious in real time, while also revealing anomalous authentication patterns and other anomalous behavior.

Organization and Best Practice

Implementing advanced analytics solutions for technology risk does not necessarily require hiring a team of expensive Ph.D.s. The democratization of data science mentioned earlier means that it is possible to achieve real results without the kind of investment in hardware, software and skills as would have been needed only a few years ago. Open-source technologies and available libraries in Python and R allow achieving measurable results with minimal technology investment. More advanced applications can be built using Apache Spark. Google,[9] Amazon[10]

> “A NEW ROLE THAT HAS RECENTLY EMERGED IS THE DATA STORYTELLER WHOSE JOB IT IS TO PROMOTE ENGAGEMENT AND ADOPTION USING CREATIVE VISUAL AND NARRATIVE JOURNEYS FOR ANALYTICAL SOLUTIONS.”

and Microsoft[11] all offer the infrastructure and tool set within their own cloud ecosystems to create machine learning products at price points affordable within the smallest of budgets.

Skills Required

However, these tools will not implement themselves; enterprises need the people and skills to do it. A successful adoption of machine-learning models at any organization requires a number of prerequisites:

- Senior management support
- Mid-level management that understands the concepts of machine learning and is convinced of the advantages it can bring
- One or more skilled people who can build, evaluate and implement data science models and tools.

Mid-level management support is extremely crucial, as these are the managers who will need to be convinced of the value of machine learning and the possibilities it can create. If they are cynical, or too wedded to traditional analytics and reporting, or see the new approaches as a threat to their own power standing within the organization, projects will drag and never achieve adoption and usage. These managers do not need to know how to code, but do need to understand conceptually what machine learning can do, what it requires as input and what kinds of problems can be solved.

Data science teams are generally organized as comprising these skill sets:

- Data scientists create and evaluate models, identify insights from data and know various algorithms that apply to different situations. Data scientists have coding skills in the data science platform being used at the organization, which is often based on Python, R, SparkML or others, including vendor-specific technologies.
- Data engineers build data pipelines, which are similar to ETL pipelines for data warehouses. Data engineers identify relevant data sources, acquire, clean, transform, join, summarize and do other activities to make data available to data scientists. In smaller organizations where workloads or use cases may be limited, data scientists may perform the data-engineering function as well. Data engineers are generally skilled in Structured Query Language (SQL), Python, Powershell or Unix scripting, connecting with APIs and other data-related technologies relevant to the organization.
- Delivery or product managers are similar to business analysts who liaise with customers, understand requirements, communicate these to the engineers and data scientists, and test the product using their domain knowledge and business context. In an IT risk organization, these are risk subject matter experts who have an understanding of the concepts, possibilities and limitations of machine learning and data science.

Larger-scale data science operations may have separate data visualization engineers whose job it is to visualize the results of data science models and other outputs using visualization tools such as QlikView, Tableau, D3 or other tools. A new role that has recently emerged is the data storyteller whose job it is to promote engagement and adoption using creative visual and narrative journeys for analytical solutions.

General Recommendations

Needless to say, committed business stakeholders are vital for the success of any data science project.

Often, larger organizations establish centralized data science teams that are expected to serve the entire organization, from marketing to human resources (HR) to risk. This is different from teams that are embedded in different functions. Sometimes the central teams only provision hardware and software, letting embedded teams use a common set of technologies. In other cases, the central teams may also hold the monopoly over hiring data scientists. In such a situation, it is up to individual function to do the job of identifying viable use cases and initiating a conversation with the central team.

Regardless of the organizational structure for data science, a number of additional points need to be kept in mind:[12]

- **Stakeholder alignment is key**—The data science team's backlog should be entirely customer-driven. What a data scientist may find interesting may not always be commercially useful. Aligning to the customers' goals is essential.
- **Getting to production**—Though at a high level one may draw parallels to the model building process with a traditional software development life cycle (SDLC), there are significant differences that make data science different from application development. Data scientists cannot operate without production data, and DevOps do not exist for modeling, meaning the traditional segregation of duties (SoD) controls, if implemented, can seriously crimp taking advantage of data science efforts.
- **Embedding in workflow**—The data science team must always be thinking of embedding their products in their customers' workflow. Stand-alone data science products can rapidly fall into disuse.
- **Managing the politics**—Technical skills are vital and necessary but insufficient by themselves for driving impact. The self-interest of the affected parties and existing organizational incentives that guide behavior need to be considered in the design of data science projects.
- **A disciplined process, not just an art**—Barring a handful of organizations operating at the bleeding edge, implementations mostly involve tried and trusted algorithms, of which many hundreds are already available to suit every possible use case. For most enterprises, the solution templates already exist and all the organizations are trying to do is to apply those templates to their situations. Machine learning projects should aim to create customer-facing applications that solve real problems and are not about mathematics.
- **Always be shipping**—Shipping, or delivering usable products, is key. The machine learning team needs to have key performance indicators (KPIs) and goals, just like any other team. Stakeholders should always be involved in setting these goals, and machine learning should be focused on real decision support, not just data that are interesting.
- **Discipline is key**—Discipline in organizational practices including performance management, idea generation, data acquisition including upstream and downstream dependency management, model selection, user acceptance, delivery, and operationalization is key to long-term success.

Conclusion

Advanced analytics and data science allows for the expansion of the reach of security metrics and technology risk measurement. Instead of merely performing post-mortem analyses of realized risk, it may be possible to get ahead of risk and prevent losses from arising by being able to predict risk scenarios on the basis of data and taking preventive action. The field is ripe for exploration and new creation. So far, the use of these technologies has been dominated by mostly commercial and direct client-facing products, e.g., fraud detection for online transactions or optimizing credit card declines. The state of the field, including software, skills and the hardware, is at a stage where the creative stakeholders can employ this at scale for managing IT risk.

> “ADVANCED ANALYTICS AND DATA SCIENCE ALLOWS FOR THE EXPANSION OF THE REACH OF SECURITY METRICS AND TECHNOLOGY RISK MEASUREMENT.”

Endnotes

1 Chollet, F.; J. J. Allaire; *Deep Learning With R*, Manning Publications, USA, 2018, *https://www.manning.com/books/deep-learning-with-r*

2 James, G.; D. Witten; T. Hastie; R. Tibshirani; *An Introduction to Statistical Learning With Applications in R*, Springer, USA, 2017, *www-bcf.usc.edu/~gareth/ISL/*

3 Lane, H.; C. Howard; H. Hapke; *Natural Language Processing in Action*, Manning Publications, USA, 2019, *https://www.manning.com/books/natural-language-processing-in-action*

4 GitHub, "Introduction to Artificial Intelligence for Security," *https://github.com/cylance/IntroductionToMachineLearningForSecurityPros*

5 GitHub, The Caret Package, *https://topepo.github.io/caret/*

6 GitHub, "Lime: Explaining the Predictions of Any Machine Learning Classifier," *https://github.com/marcotcr/lime*

7 Buczak, A.; E. Guven; "A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 2, 2016, *https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=7307098*

8 Liu, Y.; A. Sarabi; J. Zhang; P. Naghizadeh; "Cloudy With a Chance of Breach: Forecasting Cyber Security Incidents," USENIX, 12-14 August 2015, *https://www.usenix.org/system/files/conference/usenixsecurity15/sec15-paper-liu.pdf*

9 Google Cloud, "Cloud AI Products," *https://cloud.google.com/products/ai/*

10 Amazon Web Services (AWS), "Machine Learning on AWS," *https://aws.amazon.com/machine-learning/*

11 Microsoft Azure, "Machine Learning Studio," *https://azure.microsoft.com/en-us/services/machine-learning-studio/*

12 Domino, "The Practical Guide to Managing Data Science at Scale," *https://www.dominodatalab.com/resources/managing-data-science/*