Cyber GRX

A GUIDE TO

CyberGRX Data Methodology for Predictive Risk Profiles

Table of Contents

Introduction

The process of acquiring actionable cybersecurity information for a third-party cyber risk management program can be extremely time consuming, taking up to months and sometimes years to obtain assessment completion and validation. And once the initial assessment is complete, the data can quickly become stale and not relevant to emerging cyberattacks. In order to provide actionable information to influence urgent cybersecurity decisions in a reasonable time frame, CyberGRX offers a solution which provides a dynamic risk profile for any company entered into our Risk Exchange. These Predictive Risk Profiles are produced by applying advanced machine learning to data from varied sources including self-attested assessments from our third-party risk Exchange, firmographic information, and outside-in scanning data from our partners. Using this data and machine learning leveraging a graph based structure, we are able to achieve a 0.29 Hamming Loss on control predictions, control coverage group score predictions within 26.5 points on a scale of 0-100 inclusive, as well as predictions within 1±0.09 points for all maturity scores on a scale of 0-5 inclusive.

Data

CyberGRX has successfully gathered over 10,000 completed cybersecurity assessments from Exchange members. This data was further cleaned and reduced to create a set of roughly over 1,000 combined data points for training a model. The standard 80/20 split was used for creating the training and testing sets. The information in these assessments are grouped covering strategic, operational, core, management, and privacy controls of a member's cybersecurity program. The maturity level of each control family is also considered through people, process, and technology. This information is primarily binary information, with 72 being ternary and 35 being senary, confirming the existence of a cybersecurity control in place under the member's program and can range upwards of 250 in total at the CyberGRX Tier 2 Assessment level[1]. These controls are the response variables of interest in a traditional predictor-response machine learning model.

The predictor variables were chosen from a collection of external datasets with the importance of leveraging company firmographics such as industry, revenue, size, age, and online popularity for maturity and selected controls. Cybersecurity information is provided into the model through breach monitoring relating to leaked passwords, product vulnerabilities, policy violations, domain weaknesses, etc. This is paired with numerical ratings for vulnerability severity such as web security, software patching, and email security collected through automated network scanning.

TABLE 1

Feature	Data Type
Industry	Factor
Revenue	Factor
Company Size	Factor
Age	Integer
Online Popularity	Integer
Network Scanning	Float
Breach Monitoring	Factor

Table 1:
External data used as predictor variables in the model. Network scanning is a family of several variables that describe the security of a company at the domain accessible level. Breach monitoring is a family of several variables that are gathered from the dark web as well as breach signals and datasets.

[1] Tier 2 assessments allow for the confirmation of certain cybersecurity controls but do not consider the effectiveness of these controls.

Model

From a modelling perspective, we consider each assessment control a random variable that will be observed once the Exchange member has completed the assessment. With this in mind, the goal was to model the joint probability distribution of assessment answers given an Exchange member's predictor variables. As stated, the controls are primarily variables with a binomial distribution over both outcomes, with few being multinomial over three outcomes, while the maturity questions are multinomial over six outcomes. For illustration purposes, if we were to consider 200 independent binary response variables, there are $2^{200} - 1$, approximately $1.6 \cdot 10^{60}$, parameters[2] to determine for the joint distribution. Along with the sheer number of parameters to fit, the physical limitations of storing the parameters in memory is also a blocker to modelling purely off the joint distribution. Since cybersecurity controls tend to correlate with one another given the context of the question in the assessment, we can reduce the complexity of fitting a large number of parameters by leveraging conditional independence between response variables while maintaining correlations between some of them. For these reasons, a Bayesian network was utilized to model the data.

The underlying structure of a Bayesian network is a graph where each node represents a random variable and is connected to other nodes through conditional dependence, i.e. the outcome of one variable depends on one or more other variables. This graph structure is therefore directed and by design must be acyclic when constructing the conditional dependencies. This structure allows for random variables to have local distributions by only considering the variables they are dependent on, known as parent variables.

Consider two random variables ***X,Y*** in a joint probability space of possible outcomes. The chain rule of conditional probability states the probability distribution over ***X,Y***, denoted by ***P(X,Y)***, can be written as ***P(X,Y) = P(X)P(Y|X)*** where ***P(Y|X)*** is the conditional probability of ***Y given X***. This rule can then be extended to a Bayesian network structure, known as the chain rule for Bayesian networks, where for any number of variables $X_n \; for \; n = 1,2,3,\ldots$ the joint probability distribution can be written as

$$P(X_1, X_2, \ldots, X_n) = \prod_{i=1}^{n} P(X_i | \, Parents(X_i))$$

where the right hand side of the equation is the multiplication of all the conditional probabilities from 1 to ***n*** in the graph for a potential outcome only considering the parents of each variable. When a variable has no parents, it is simply the marginal distribution $P(X_i)$. This factorization of the joint probability distribution allows for the parameter space to reduce in size to at most $n \cdot 2^k$ where ***n*** is the number of variables and ***k*** is the largest number of parents for a variable. Usually, variables only depend on a very small number of parents for their conditional probabilities, making the previous exponential parameter space linear in the variables.

In addition to parameter reduction, Bayesian networks have several properties that are advantageous to modelling assessment data. First, due to the statistical nature of the model, they are explainable compared to other popular models that fit high dimensional data. Expert domain knowledge can also be included as prior distributions over the response variables. When performing inference, the model still produces predictions for the response variables even if there is missing data in the input. Finally, the theory behind Bayesian networks has been developed over decades and their application to datasets with large amounts of binary variables is well known.

[2] The 1 is subtracted from the exponential term since the last probability is fully determined by all the probabilities before it.

Training

Training a Bayesian network is a two step process: learning the structure of the dependencies, and fitting the parameters of the variables.

There are several algorithms available to learn the structure of the network which categorize into three types: constraint-based, score-based, and hybrid algorithms. In this development we used a popular score-based algorithm called the hill-climbing (HC) algorithm. HC finds a local optimum by searching the possible orientations of edges connecting one variable to another and assigning a score to the potential structure. The Bayesian Information Criterion (BIC) was used to score the structures, defined as:

$$BIC = p \cdot ln(n) - 2 \cdot ln(P(X|\hat{\theta}, M))$$

where ***ln(•)*** is the natural logarithm, ***p*** is the number of parameters in the model and ***n*** is the number of observations in the data. $P(X|\hat{\theta}, M)$, known as the likelihood function, is the probability of observing the data ***X*** given the estimated parameter values $\hat{\theta}$ and the model ***M***, i.e. the parameters that fit the data the best. The BIC score penalizes large amounts of parameters to the model while maximizing the likelihood function. Penalizing large parameters prevents the complexity of the model from getting too large which may result in overfitting the training data leading to poor performance on unseen data. The HC algorithm will iteratively search for a maximum BIC score until either the pre-defined maximum number of iterations is reached or increases to the score are no longer found.

When searching for a structure that describes the (in)dependencies of the underlying distribution, a challenge that needed to be overcome is independence equivalence, or I-equivalence. I-equivalence is the property that two structures with different directed edges still encode the conditional independencies of the underlying distribution given the data. Since two different graphs can encode the same conditional independencies of the variables, there is nothing to say that one is better than the other if the only thing different is the direction of the edges. One way to mitigate this is to define edges before starting the structure search. With the assistance of cybersecurity professionals at CyberGRX, several of the input variables were manually mapped to assessment controls in order to force dependencies between variables and reduce the search space for the structure. Examples of these mappings are in Figure 1 where several of the inputs were mapped to selected variables in the graph.

FIGURE 1

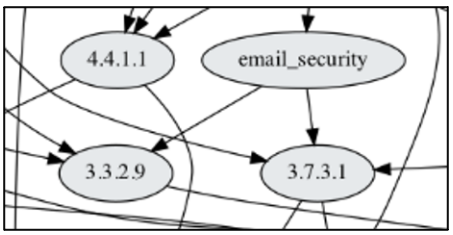


Figure 1.1: Email security input mapped to two controls

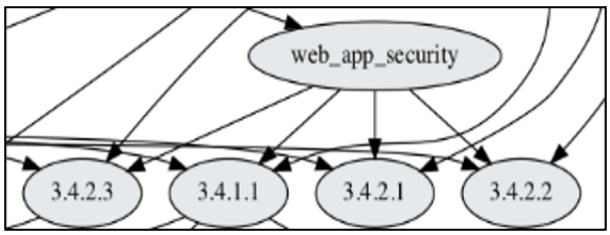


Figure 1.2: web app security input mapped to four controls

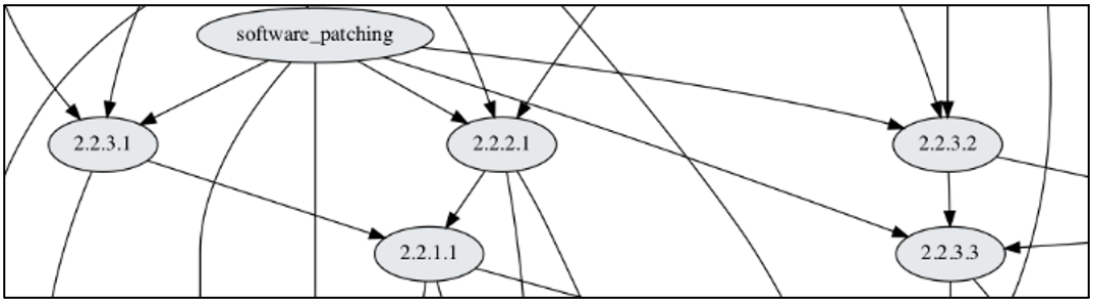


Figure 1.3: software patching input mapped to six controls; four included in the image.

Figure 1:

Screenshots from the structure of the model where the input nodes were manually mapped.

Once the structure is found, the conditional dependencies between the variables are now known and we can fit the parameters of the variables in the graph using the training data. The probabilities of the outcomes are stored in a Conditional Probability Distribution (CPD) table that captures the local distribution of a variable given its parents in a matrix of dimension

$$\left|X_i\right| \times \prod_{j=1}^{k} \left|Parent_j(X_i)\right| \; for \; i = 1, 2, 3...$$

where the |•| denotes the number of outcomes, or cardinality, for the variable between the pipes.

A prior probability distribution was included over the outcomes of the variables to smooth out the bias in the assessment answers. The Bayesian Dirichlet equivalent uniform (BDeu) distribution was used for two reasons; it makes computation of the distribution easier and it assumes a uniform distribution over the outcomes a priori. Before fitting a variable in the network with the data, we use a standard count for the prior distribution that is uniform across all outcomes of the CPD. The counts assigned to the outcomes are calculated from the equation

$$\alpha_{ij} = \frac{q}{|X| \times \prod_{j=1}^{k} |Parent_j(X)|}$$

where α_{ij} is the value we will set to the ith row and jth column entries of the CPD, ***q*** is the equivalent sample size hyperparameter, and the denominator in the equation is simply the number of entries the CPD table has. Since all the α_{ij} will be equivalent, this prior distribution is considered a uniform distribution across the CPD.

Since the Dirichlet prior is a conjugate prior, adjusting the probabilities is a relatively simple procedure. A conjugate prior is a distribution that will be the same distribution after including information from observations. To fit the probability of an outcome in the CPD using the data, we fix the outcome of the parents and add the normalized frequency of the variable's outcome to the α_{ij}. An example of a fitted CPD table is shown in **Table 2** where the columns are the conditional distributions that depend on the outcome of the parents.

TABLE 2

Parent 1	Parent 1 (0)	Parent 1 (0)	Parent 1 (1)	Parent 1 (1)
Parent 2	**Parent 2 (0)**	**Parent 2 (1)**	**Parent 2 (0)**	**Parent 2(1)**
Variable (0)	0.86	0.25	0.4	0.7
Variable (1)	0.14	0.75	0.6	0.3

Table 2:

An example of a CPD table where the columns are the conditional probabilities of the variable given the observed outcomes of the parents shown in parenthesis.

Querying

In order to make predictions to the assessment answers given a company's input data, or evidence, we must query the Bayesian network for inference on those variables. Several methods exist for generating outcomes on the response variables. These include exact inference methods to compute the joint probability given the structure of the graph, particle based methods that generate data points given the evidence, and maximum a posteriori (MAP) queries which return outcomes that maximize the probability of observing the evidence. We use particle based methods in the form of random sampling from the Bayesian network to produce more robust analysis on the potential assessment outcomes. Given enough samples, we can approximate the true distribution. In particular, likelihood-weighted sampling was used for the approximation.

Likelihood-weighted sampling (LW) provides the ability to fix the input variables and adjust the CPDs of the random variables in order to fit the event of observing the evidence. LW is a subset of a larger form of sampling called Importance Sampling where the topological ordering[3] of the graph determines the order of importance to sample the variables. Sampling a variable consists of using the CPD column with the parent outcomes fixed to be the observations of the evidence and then drawing an outcome from that CPD with probabilities corresponding to those of the selected columns. The results of the sampled variables, being parents, in that order feed into the CPDs of the variables that depend on them, known as children, until the end of the ordering is reached. This process is then repeated for a predefined number of times to produce the same number of potential assessments.

The weighting strategy in LW stems from rejection sampling where the Bayesian network is sampled in the topological ordering without evidence. The samples would then be rejected if they did not match the evidence observations. This can become quite inefficient if the probability of observing the evidence is low. In LW, the weighting adjusts the significance of the sample to reflect the likelihood of the observed evidence's probability given its parents.

In order to process a company for results, it must have a set of possible assessments to analyze given the observed predictor variables for that company. A sample size of 1000 was chosen for LW since it was both performative in time to completion as well as accuracy in approximation. It takes roughly 30 seconds to score and analyze a block of 1000 sampled assessments for a company. This computation is then distributed to run in parallel for multiple companies to obtain results.

[3] Topological ordering refers to listing the nodes in a directed graph such that nodes that have parents or children are not listed before their parents and are not listed after their children.

Predictive Results Calculations

In order to provide insights into a company's cybersecurity program, it is necessary to remain consistent with the already established metrics and summaries provided from self-assessed assessments. When an Exchange member at CyberGRX completes an assessment, scores are generated from their answers to provide an overview of the level of risk in their cybersecurity program. These include coverage of the five control groups listed at the beginning of this document, maturity group coverage, and a gaps analysis to identify weaknesses through the MITRE ATT&CK® framework. To estimate the true scores, we chose to take the approach of describing the possible distribution of scores for a company by querying the possible assessments from the Bayesian network thus producing what we call a sample block. An illustration for clarity of the sample block is included in **Table 3**. This provides a more robust view of the possibilities along with building a confidence score around the distribution of scores.

TABLE 3

Outcomes	Question 1	Question 2	...	Question 251
Assessment 1	Yes	Yes	...	No
Assessment 2	Yes	Yes	...	Yes
...	...	...	...	...
Assessment 1000	No	Yes	...	No

Table 3:
A sample block is produced for each individual company with 1000 possible assessments in the block.

The process to produce coverage and maturity scores at the group level begins by iterating through each assessment in the sample block and scoring that possible assessment for each group. Once all the sampled assessments have been scored, a histogram per group is created with the scores. Having a histogram allows us to display the median as the expected value of the scores and a confidence[4] around the expected value for the control coverage.

The maturity coverage is displayed as a range of scores: low, median, and high. Where the median is the expected value of the score, and the low and high scores cover a margin of error for the estimate.

Each sampled assessment produces a list of ranked gaps from the MITRE ATT&CK® analysis. Each question is then scored by a point accumulation system based on its position in each list. The accumulation of these points over all lists is then used to rerank the union of all the gaps outputs in order to produce the top five across sampled assessments with an accompanying confidence[5] score.

[4] The confidence for the coverage scores is the probability that the true score will not deviate more than a proprietary margin of error from the expected value.

[5] The confidence for each of the five gaps is calculated by their probability of not being in place within the set of all sampled assessments.

Performance Results

Performance results for the model were separated into coverage scoring results and the maturity scoring results. Although the coverage can have Not Applicable outcomes, the modelling problem is a multi-label problem. On the other hand, the modelling problem for the maturity is considered multi-class and multi-label. The outcomes range from 0-5 making it multi-class and there are 28 different maturity questions making it multi-label thus they cannot be considered together.

Coverage predictions were scored using the Mean Absolute Error (MAE) which describes the average distance of the errors from the actual scores. To calculate the MAE we used the equation

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |y_i - \widehat{y_i}|$$

where ***n*** is the size of the test set, ***y*** is the true group score for an assessment and $\hat{y}$ is the predicted score. The right hand side with the $\sum_{i=1}^{n}$ symbol states the summation of all the distances in the samples from the true score. The $|\bullet|$ in this case denote taking the absolute value of the difference. This only considers how far an estimate was from the truth without having effects from the sign of that difference. The value after performing the summation is then normalized using the size of the test set by multiplying ***1/n***. The MAE for each coverage group is shown in the first column of **Table 4**.

Since each prediction of coverage is accompanied by a confidence level, we measured how often that confidence is correct in indicating whether or not the true score is captured within a proprietary margin of error[6]. A cutoff of 50% on the probability of capturing the true score within the margin of error was used. This was calculated using the equation

$$Confidence\ Precision = \frac{1}{n}\sum_{i=1}^{n} 1_{\{|y_i - \widehat{y_i}| < \epsilon\,:\, c \geq 0.5\}}$$

where ϵ is the proprietary margin of error for the estimate and ***c*** is the confidence that the estimate is contained within that margin. The $1_{\{|y_i - \hat{y}_i| < \epsilon:\, c \geq 0.5\}}$ indicates to add 1 to the summation if the true score is within the margin of error and the confidence associated with capturing the true score was at least 50% and if otherwise to add 0. The results are shown in the second column of **Table 4**.

[6] The margin of error is considered a hyperparameter in this work and therefore different models will require a different margin.

Since predicting coverage is a binary multi-label classification problem, we can use the Hamming Loss function to score our model for performance on accuracy. The Hamming Loss reflects the number of labels predicted wrong over the total number of labels. Since the Hamming Loss measures the mislabelling rate, a lower value for the Hamming loss is preferred over a higher one. The metric ranges in values between 0-1 and is calculated as follows.

$$Hamming\ Loss\ = \frac{1}{|N|\cdot|L|}\sum_{i=1}^{|N|}\sum_{j=1}^{|L|} 1_{\{y_{ij} \neq z_{ij}\}}$$

Where ***|N|,|L|*** are the number of samples and the number of labels we are trying to predict respectively. The values y_{ij}, z_{ij} are the true label and predicted label for the ***ith*** assessment on the ***jth*** control. The $1_{\{y_{ij} \neq z_{ij}\}}$ states we will add 1 if we missed the label, i.e $y_{ij} \neq z_{ij}$, and 0 otherwise. The Hamming Loss was calculated per group as well as over all groups and is shown in the third column of **Table 4.**

TABLE 4

	Mean Absolute Error [0-100]	Confidence Precision [0-1]	Hamming Loss [1-0]
Strategic	21.31	0.96	0.24
Core	27.15	1.0	0.28
Operational	16.77	1.0	0.3
Management	22.06	0.95	0.29
All Groups Above	-	-	0.29
Privacy	2.5	1.0	0.025

Table 4:

Mean absolute error on a scale of 0-100, confidence precision of the true score per group within the margin of error given confidence of 50% or more, and hamming loss.

For measuring the performance of the maturity question predictions, we used the MAE to see how far the estimate was from the true value as well as the true value being captured within the margin of error for each group. Note the maturity questions range between 0 and 5, instead of 0-100, which is why the MAE are much lower compared to the controls coverage. **Table 5** shows the results of these metrics.

TABLE 5

	Mean Absolute Error [0-5]	Confidence Precision [0-1]
Strategic Maturity	0.96	0.41
Core Maturity	0.97	0.37
Operational Maturity	0.9	0.39
Managegment Maturity	1.06	0.36
Privacy Maturity	1.08	0.4

Table 5:

Mean absolute error and confidence precision of the true score within the margin of error at 50% confidence. Maturity is scored on a scale of 0-5.

Since residual risk is an output of the scoring, we were able to measure the predicted overall residual risk compared to the true overall residual risk. The residual risk is a way to quantify the reduction in risk from having certain cybersecurity controls in place, thus reducing the inherent risk of the threat landscape to a company. We use the MAE as a measure of how far we are from estimating the residual risk from the true residual risk in the test set. The Confidence Precision is also used to see how often the true residual risk is captured within the margin of error. An important measurement that we have not discussed yet is the Lower Bound Cutoff which is a way to measure how often the true overall residual risk is kept above the predicted lower bound. This is important to note because the lower the overall residual risk, the less risk a company poses to their customer(s) and not keeping the true overall residual risk above a lower bound dilutes the validity of the prediction. The results are shown in **Table 6.**

TABLE 6

	MAE [0-100]	Confidence Precision [0-1]	Lower Bound Cutoff [1-0]
Overall Residual Risk	8.83	0.33	0.71

Table 6:

The overall residual risk measured through MAE, high confidence precision, and Lower Bound Indication

By design of the system, there will always be five predicted gaps outputs with an accompanying confidence level. The set of five predicted gaps can contain both high gaps and low gaps[7]. Analytics from scoring an attested assessment can output zero or more gaps. For this reason we measured the performance of the predicted gaps primarily through the intersection of the predicted set and the attested set. If the predicted set intersects at all with a non-empty attested set then that will contribute to the score. **Table 7** has the rate of intersection for the gaps. As shown, 43% of the time do predicted gaps intersect with the attested gaps. Predicted high gaps, or gaps with a confidence of 40% or more, intersect with the attested gaps 26% of the time.

TABLE 7

	Gap Intersection Rate [0-1]	High Gap Intersection Rate [0-1]
Predicted Gaps	0.43	0.26

Table 7:

Predictive analytics will always output five gaps regardless of the attested assessment. The gap intersection rate is a measure of the intersection between predicted and attested gaps. The high gap intersection rate is similar with the condition of only predicted gaps with an accompanying confidence level of ≥40% are considered.

[7] High gaps are missing controls which pose a high level of risk from MITRE ATT&CK® techniques. Low gaps are controls that are in place but still pose a high level of risk without understanding their implementation effectiveness.

Future Work

Throughout the development of this model we identified several areas of improvement. The first is centering on a structure that has the most information propagation without overwhelming variables with an excessive amount of parents. Although the HC algorithm allows us to search for a structure that optimizes the BIC score, further reducing the search space is necessary.

More inputs with high amounts of predictive power would adjust the imbalance in the predictors to response ratio. Continuously exploring and expanding the input space would allow to reduce the error in predicting an assessment.

Different sampling methods are of interest to experiment with. There are several algorithms that yield less error in less time per sample than likelihood weighted sampling.

Addendum

Model v1.1 Release Notes

Model v1.1 was released June 24, 2022.

PERFORMANCE CHANGES

Average F1 Score of 0.3
Note: F1 is a measurement of true positive and false positive rates. The closer to 1.0 we are, the less we have.

F1% improvement from previous model

- Median: 6%
- Max: 83%

Assessment Accuracy

- Lowest quartile (25th) is 58% accurate.
- Median is 68% accurate.
- Model is up to 91% accurate.

Sub-control Accuracy

- 25% of sub-controls are 80% accurate.
- Half of the sub-controls are 75% accurate.
- We are up to 95% accurate on specific sub-controls.

TABLE 8

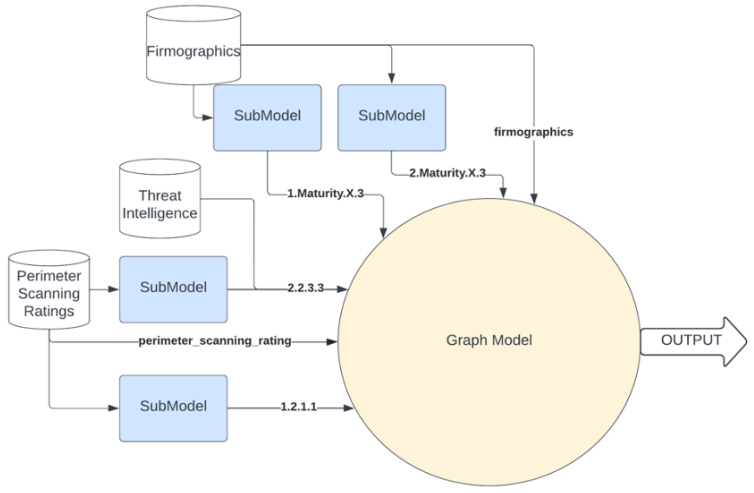
Hamming Loss	Group 1	Group 2	Group 3	Group 4	Overall
v1.0	0.14	0.28	0.2	0.22	0.21
v1.1	0.14	0.37	0.23	0.24	0.24

Recall Hamming Loss measures the mislabelling rate, a lower value for the Hamming loss is preferred over a higher one.

Upon deeper analysis, we found that our external data feed scores dropped over time for multiple companies which pulled down our coverage predictions, subsequently scoring them lower than their historical attested results. This led to a larger Hamming loss score.

Architecture Changes

Decision Trees were trained to improve predictions for perimeter scanning mapped subcontrols and Random Forest models for maturity question predictions. These helper models, or SubModels, are tuned to split the perimeter scanning ratings using an internal decision function for minimizing the mislabelling probability. The SubModel predictions are then passed to the graph model to initialize the internal sampling for inference.



This strategy improves development time by reducing the data preprocessing to find the best thresholds to the continuous features. In this way, 82 assessment question predictions were improved.

Cyber GRX

Questions? Reach out to your CyberGRX Account Manager for further information