# Uses and misuses of Bayes' rule and Bayesian classifiers in cybersecurity

Gregory V. Bard

Citation: AIP Conference Proceedings **1910**, 020008 (2017); doi: 10.1063/1.5013945
View online: https://doi.org/10.1063/1.5013945
View Table of Contents: http://aip.scitation.org/toc/apc/1910/1
Published by the American Institute of Physics

# Uses and Misuses of Bayes' Rule and Bayesian Classifiers in Cybersecurity

Gregory V. Bard

*Dept. of Math., Stat., and Comp. Sci., Jarvis Hall Science Wing,*

*The University of Wisconsin—Stout, Menomonie, Wisconsin, 54751, USA*

bardg@uwstout.edu

**Abstract.** This paper will discuss the applications of Bayes' Rule and Bayesian Classifiers in Cybersecurity. While the most elementary form of Bayes' rule occurs in undergraduate coursework, there are more complicated forms as well. As an extended example, Bayesian spam filtering is explored, and is in many ways the most triumphant accomplishment of Bayesian reasoning in computer science, as nearly everyone with an email address has a spam folder.

Bayesian Classifiers have also been responsible significant cybersecurity research results; yet, because they are not part of the standard curriculum, few in the mathematics or information-technology communities have seen the exact definitions, requirements, and proofs that comprise the subject. Moreover, numerous errors have been made by researchers (described in this paper), due to some mathematical misunderstandings dealing with conditional independence, or other badly chosen assumptions.

Finally, to provide instructors and researchers with real-world examples, 25 published cybersecurity papers that use Bayesian reasoning are given, with 2–4 sentence summaries of the focus and contributions of each paper.

## 1 Overview

The word Bayesian occurs frequently in the titles of papers (e.g. consult the bibliography of this paper). While the reader has probably worked with the most elementary form of Bayes' rule during undergraduate coursework, there are more advanced forms and applications. A highly related, though far more complicated object, is a Bayesian Classifier. While not discussed at all in any undergraduate textbooks that the author could find, Bayesian Classifiers are already very popular in cybersecurity research. Moreover, the author has encountered serious mathematical misunderstandings that have caused scholars at multiple levels to make incorrect statements. Mostly, these misunderstandings relate to conditional independence, a topic that will be explored at length in this paper.

Section 2 reviews two forms of Bayes' Rule that we will need in the rest of the paper. Appendix A, however, lists all ten forms in detail—as well as some background about why they are different in practice—even if they all can be derived from what this paper calls "the universal form." We also argue that because textbooks give only the elementary form, and there are many forms needed in practice, there is an intellectual gap which disadvantages researchers.

Section 3 presents the spam formula. Bayesian Spam Filtering is in some ways the major triumph of Bayesian research in cybersecurity, because everyone who has an email address today has a spam folder. Perhaps if spam filtering were not so advanced, email would have fallen out of use long ago, much like bulletin boards (BBS's) and usenet newsgroups did. We show the derivation and pay careful attention to the assumptions, as well as discuss the consequences of the most bizarre assumption, that spam versus non-spam emails occur with probability $1/2$.

Section 4 presents the Bayesian Classifier formula and algorithm. The more mathematical specifics are placed in the appendices, so various readers can decide for themselves what they need to read. The derivation of the Bayesian Classifier Formula, with careful attention being given to the assumptions, is in Appendix B. The appendices of this paper can be found at `www.gregorybard.com` after clicking on "publications."

Appendix C shows that the Bayesian Classifier formula, restricted to $n = 1$, becomes the original Bayes' Rule. In particular, it turns into the finite-predicts-finite form. Appendix D shows that the spam formula is just the Bayesian Classifier restricted to $n = 2$ and $m = 2$, subject to a few other assumptions.

*Proceedings of the 43rd International Conference Applications of Mathematics in Engineering and Economics*
AIP Conf. Proc. 1910, 020008-1–020008-8; https://doi.org/10.1063/1.5013945
Published by AIP Publishing. 978-0-7354-1602-4/$30.00

Both the spam formula and Bayesian Classifiers have as part of their assumptions some conditional independence statements. Because conditional independence is an advanced topic, some researchers unfamiliar with it might desire to avoid it. For the spam formula, we show a numerical counter-example in Appendix E. The counter-example satisfies independence, but not conditional independence, and results in the wrong answer being obtained. Therefore, conditional independence is not a concept that can be dispensed with. Since the spam formula is just a Bayesian Classifier restricted to $n = 2$ and $m = 2$, the necessity also applies to Bayesian Classifiers.

Appendix F will demonstrate that the "proof" of the spam formula, as presented in the famous encyclopedia *Discrete Mathematics and Its Applications* [16], by Kenneth Rosen, is actually incorrect. This will be done with an explicit numerical counter-example, which meets Rosen's assumptions, but causes incorrect answers to be obtained. (Rosen did not correctly state the conditional independence requirements, a matter which will be described in that appendix.)

Appendix H presents 25 papers, each applying some aspect of Bayesian methods to important issues in cybersecurity. Rather than a mere list of bibliographical citations, two to four sentences describing each paper's contributions and focus are given.

## 2 Two Crucial Forms of Bayes' Rule

In Appendix A, we discuss the many forms that Bayes' Rule can take. We list ten forms there, but one could easily argue for five or for seventeen. (These are not arbitrary numbers. Note, $3^2 + 1 = 10$, $2^2 + 1 = 5$, and $4^2 + 1 = 17$.) It is mathematically true that all of those forms can be derived from the "universal form." However, researchers wanting to analyze data in applications might not desire to go through the error-prone and relatively difficult task of such derivations. Looking through all of those forms is a useful review, and it demonstrates the subtleties of Bayesian reasoning. However, for this paper, we only need two forms. The first is the boolean-predicts-boolean form:

$$\begin{aligned}
Pr[B|A] &= \frac{Pr[B]Pr[A|B]}{Pr[B]Pr[A|B] + Pr[B^c]Pr[A|B^c]} \\
Pr[B^c|A] &= \frac{Pr[B^c]Pr[A|B^c]}{Pr[B]Pr[A|B] + Pr[B^c]Pr[A|B^c]} \\
Pr[B|A^c] &= \frac{Pr[B]Pr[A^c|B]}{Pr[B]Pr[A^c|B] + Pr[B^c]Pr[A^c|B^c]} \\
Pr[B^c|A^c] &= \frac{Pr[B^c]Pr[A^c|B^c]}{Pr[B]Pr[A^c|B] + Pr[B^c]Pr[A^c|B^c]}
\end{aligned}$$

The second one is the finite-predicts-finite form:

$$\text{for all } B_i \in \{B_1, B_2, \ldots, B_m\}, \text{ and for all } A_k \in \{A_1, A_2, \ldots, A_n\}, \quad Pr[B_i|A_k] = \frac{Pr[B_i]Pr[A_k|B_i]}{\sum_{j=1}^{j=m} Pr[B_j]Pr[A_k|B_j]}$$

Of course, the names refer to the fact that $A$ and $B$ are boolean random variables in the first formula, and are elements of a finite set in the second formula. The reader is encouraged to review the full exploration in Appendix A.

## 3 Critical Analysis of the Spam Formula

As a case study for exploring the role of excessive simplifying assumptions, and their impact on the uses of Bayesian reasoning in cybersecurity, let us consider "the spam formula," which is widely regarded in computer science. Specifically, let us consider the presentation of it in Kenneth Rosen's encyclopedia *Discrete Mathematics and Its Applications* [16], a textbook of fundamental importance in American computer science education. In the 7th edition, "the spam formula" is in Section 7.3, "Bayes' Theorem" where it serves as Example 4, but with the assumptions incorrectly stated. The (incorrect) proof of the formula is Exercise 23, which is solved in the back of that textbook. It is worth mentioning that Bayesian Spam Filtering occupies approximately three out of seven pages of that section—this is not a minor nor obscure application of Bayes' Rule.

As it turns out, while the formula is popular and frequently quoted, it is based on three assumptions. As you might guess, if the assumptions are false in practice, the formula produces wrong answers. However, the assumptions are

very subtle, and we will show that Kenneth Rosen's encyclopedia has the assumptions incorrectly, which invalidates his derivation of the formula. Appendix F will demonstrate this with an explicit counter-example, and show it causes incorrect answers to be obtained.

## 3.1 The Spam Formula Itself

Consider a pair of words, Word 1 and Word 2, which are likely to occur in spam emails and unlikely to occur in non-spam emails. Let $p_1$ and $p_2$ be the probability of finding Word 1 and Word 2, respectively, in a spam email. Let $q_1$ and $q_2$ be the probability of finding Word 1 and Word 2, respectively, in a non-spam email. The spam formula claims that the probability that an email is spam, given that it contains both Word 1 and Word 2, is given by

$$\frac{p_1 p_2}{p_1 p_2 + q_1 q_2}$$

Armed with many such pairs of words, one can build a "Bayesian Spam Filter." Based on a collection of emails (called a corpus) which have been carefully sorted into "spam" and "non-spam" subsets, a search will be made for word pairs that cause the formula above to output a number greater than 0.95, or some other arbitrary cutoff. Once the list of word-pairs is finalized, any email containing one (or more) of those pairs will be marked as spam. Only emails containing none of those pairs will be marked as non-spam. This is a common technique for spam filtering, frequently used by internet service providers (ISPs). Alternatively, triples of words can be used, and the formula becomes $p_1 p_2 p_3/(p_1 p_2 p_3 + q_1 q_2 q_3)$. We will restrict to word pairs in this paper, because the analysis of the triples would be very similar.

We will now explore the underlying assumptions needed to derive this formula from Bayes' Rule, and then proceed to explore the consequences of those assumptions.

## 3.2 Necessary Notation and Assumptions for the Spam Formula

Let $S$ be the event that a particular email is spam. Let $W_1$ be the event that Word 1 occurs somewhere in the email, and let $W_2$ be the event that Word 2 occurs somewhere in the email.

In the derivation presented in this paper, there are three assumptions. First, we assume that there is an equal probability that an email is spam or non-spam, which of course is completely arbitrary. The author would prefer not to make such an assumption, and it is easy to show that if we did not make it, then the spam formula would become

$$\frac{Pr[S]p_1 p_2}{Pr[S]p_1 p_2 + Pr[S^c]q_1 q_2}$$

instead. (This occurs as the second-to-last equation in Appendix D.) Nonetheless, since the spam formula is so widely quoted, the author does not feel as though it is right to edit the formula.

Second, we assume that the events $W_1|S$ and $W_2|S$ are conditionally independent. This assumption means

$$Pr[(W_1 \cap W_2)|S] = Pr[W_1|S]Pr[W_2|S]$$

Third, we assume that the events $W_1|S^c$ and $W_2|S^c$ are conditionally independent. This assumption means

$$Pr[(W_1 \cap W_2)|S^c] = Pr[W_1|S^c]Pr[W_2|S^c]$$

Since this paper makes use of conditional independence, a concept not familiar to all students, it is also worth showing that the formula fails miserably even when $Pr[S] = 1/2$ is true and $W_1$ is independent of $W_2$, but without the conditional independence assumptions. Therefore, conditional independence is not a concept that can be dispensed with. That is thoroughly worked out in Appendix E.

In his proof of the spam formula, Rosen presents assumptions that are different. What Rosen states is that $Pr[S] = 1/2$, and $W_1$ and $W_2$ are independent events, as well as that the events $W_1|S$ and $W_2|S$ are conditionally independent. One might easily imagine that the independence of $W_1$ and $W_2$ as well as the conditional independence of $W_1|S$ and $W_2|S$ might together imply that the events $W_1|S^c$ and $W_2|S^c$ are conditionally independent. The author of this paper himself was mistakenly under this impression until corrected (with a counter-example) by Prof. Jing Xi of the University of Wisconsin—Stout's Department of Mathematics, Statistics, and Computer Science.

In Appendix F, this paper will show that the independence of $W_1$ and $W_2$ as well as the conditional independence of $W_1|S$ and $W_2|S$ do not together imply that the events $W_1|S^c$ and $W_2|S^c$ are conditionally independent. Moreover, it is shown there that the spam formula can produce significantly wrong answers when $W_1|S^c$ and $W_2|S^c$ are not conditionally independent.

A reader who wishes to compare the work of this paper with Rosen should be aware that he uses the words "we have no prior knowledge regarding whether or not the message is spam" to indicate that $Pr[S] = 1/2$. This is in Exercise 23, which is Rosen's proof[1] for the spam formula. That wording is extremely strange, and it would be much better to simply say $Pr[S] = 1/2$.

## 3.3 The Derivation of the Spam Formula

(This derivation is different from the one given in Rosen's encyclopedia, in order to highlight the assumptions and the connection with Bayes' Rule.) We start with the boolean-predicts-boolean form of Bayes' Rule,

$$Pr[B|A] = \frac{Pr[B]Pr[A|B]}{Pr[B]Pr[A|B] + Pr[B^c]Pr[A|B^c]}$$

and then substitute $B = S$ (the event that an email is spam), and $A = W_1 \cap W_2$, the event that some email contains both Word 1 and Word 2. Making this substitution, we obtain

$$Pr[S|(W_1 \cap W_2)] = \frac{Pr[S]Pr[(W_1 \cap W_2)|S]}{Pr[S]Pr[(W_1 \cap W_2)|S] + Pr[S^c]Pr[(W_1 \cap W_2)|S^c]}$$

If we make the first assumption, that $Pr[S] = 1/2$. Also, $Pr[S^c] = 1 - Pr[S] = 1/2$. Substituting these, we get

$$Pr[S|(W_1 \cap W_2)] = \frac{(1/2)Pr[(W_1 \cap W_2)|S]}{(1/2)Pr[(W_1 \cap W_2)|S] + (1/2)Pr[(W_1 \cap W_2)|S^c]} = \frac{Pr[(W_1 \cap W_2)|S]}{Pr[(W_1 \cap W_2)|S] + Pr[(W_1 \cap W_2)|S^c]}$$

Recall,

$$p_1 = Pr[W_1|S] \qquad p_2 = Pr[W_2|S] \qquad q_1 = Pr[W_1|S^c] \qquad q_2 = Pr[W_2|S^c]$$

Because $p_1p_2 = Pr[W_1|S]Pr[W_2|S]$ and $q_1q_2 = Pr[W_1|S^c]Pr[W_2|S^c]$, in order to reach Rosen's result, we must assume that

$$Pr[(W_1 \cap W_2)|S] = Pr[W_1|S]Pr[W_2|S] \quad \text{as well as} \quad Pr[(W_1 \cap W_2)|S^c] = Pr[W_1|S^c]Pr[W_2|S^c]$$

demonstrating explicitly the reliance on conditional independence.

If we do make those two highly-related assumptions, we obtain

$$\begin{aligned} Pr[S|(W_1 \cap W_2)] &= \frac{Pr[(W_1 \cap W_2)|S]}{Pr[(W_1 \cap W_2)|S] + Pr[(W_1 \cap W_2)|S^c]} \\ &= \frac{Pr[W_1|S]Pr[W_2|S]}{Pr[W_1|S]Pr[W_2|S] + Pr[W_1|S^c]Pr[W_2|S^c]} \\ &= \frac{p_1p_2}{p_1p_2 + q_1q_2} \text{ as desired.} \end{aligned}$$

## 3.4 A Critique of the First Assumption

The first assumption, $Pr[S] = 1/2$, seems arbitrary. The relative error of that assumption is the relative error of using

$$\frac{p_1p_2}{p_1p_2 + q_1q_2} \text{ instead of } \frac{Pr[S]p_1p_2}{Pr[S]p_1p_2 + Pr[S^c]q_1q_2}$$

[1] One cannot help but wonder at why this important derivation is presented as a homework exercise. Clearly, it is too difficult, as Kenneth Rosen has gotten it wrong himself.

More generally, that equals the relative error of using

$$Pr[B|A] = \frac{Pr[A|B]}{Pr[A|B] + Pr[A|B^c]} \text{ instead of } Pr[B|A] = \frac{Pr[B]Pr[A|B]}{Pr[B]Pr[A|B] + Pr[B^c]Pr[A|B^c]}$$

as the boolean-predicts-boolean case of Bayes' Rule. One can easily compute that this is

$$\frac{Pr[A|B^c]}{Pr[A|B] + Pr[A|B^c]} \left( \frac{Pr[B^c]}{Pr[B]} - 1 \right)$$

or for the spam formula

$$\frac{q_1 q_2}{p_1 p_2 + q_1 q_2} \left( \frac{Pr[S^c]}{Pr[S]} - 1 \right)$$

and unsurprisingly, this comes to zero when $Pr[S] = Pr[S^c] = 1/2$ is actually true.

For spam filtering, consider the realistic of $Pr[S] = 1/6$ and $Pr[S^c] = 5/6$, with $p_1 = 0.91$, $p_2 = 0.93$, $q_1 = 0.11$, and $q_2 = 0.09$. We see that the term in the parentheses will be 4. The uncorrected spam formula gives an answer of $0.988437281\cdots$, where as the true answer is $0.944742129\cdots$. This is a relative error of about 4.63%. Admittedly, this is not much of a problem—for spam filtering.

### 3.5 A Hypothetical Application: Anti-terrorism Operations

Suppose some intelligence agency or police force wants to search for text messages relating to plans for terrorist attacks. There could be words that are in common use by someone planning such attacks, that are rare in other circumstances. The spam formula can easily be used for such a purpose, as can Bayesian Classifiers. Because Bayesian Classifiers reduce to the spam formula when $n = 2$ and $m = 2$ (as shown in Appendix D), it makes sense to consider this example. Let $Pr[T] = 1/1000$ be the probability that a random text message, of all of the text messages sent in a particular city or province, is related to terrorism. Using the $p_1$, $p_2$, $q_1$, and $q_2$ from the previous paragraph, we obtain a relative error of 1153.96%, because the multiplier in the parentheses is 998. The uncorrected formula gives an answer of $0.988437281\cdots$, as before, but the true probability is $0.008476868\cdots$.

It has been widely reported in the newspapers that such monitoring of text messages and social media is taking place, but that the police and intelligence services cannot handle the large number of people who end up on "the watch lists." This is a clear indication of a large number of false positives, making "the watch lists" unnecessarily large.

The paper [3] made a somewhat similar observation, but the context of network intrusion detection systems. A main point of that paper is that misestimates of $Pr[B]$, sometimes called "the base rate," can be responsible for a large number of false positives. Simply assuming $Pr[B] = 1/2$ would be very unwise when $Pr[B]$ is actually much smaller.

## 4 About Bayesian Classifiers

Because cybersecurity is composed of many disciplines (e.g. mathematicians working on cryptography, computer scientists working on operating systems, computer engineers working on authentication hardware, information technologists working on router configurations, etc…), it is safe to say that the majority are not familiar with the precise definitions, formulas, assumptions, and derivations of Bayesian Classifiers, though many have heard the phrase. The goal of this paper (with its appendices) is to spell it out, in as much detail as possible, pointing out the assumptions along the way.

### 4.1 An Overview of Bayesian Classifiers

The goal of a Bayesian Classifier is to examine a set of objects, and place each into one of $m$ classes $C = \{c_1, c_2, c_3, \ldots, c_m\}$. Every object will be examined in terms of $n$ attributes, which are random variables, comprising $\vec{X} = (X_1, X_2, X_3, \ldots, X_n)$. For any particular object, the values measured for each attribute are denoted $\vec{x} = (x_1, x_2, x_3, \ldots, x_n)$.

The random variables $X_i$ could be boolean, or they could be selected from a finite set. They could also be non-negative integers, mapped into a finite set such as $\{0, 1, 2, 3, \ldots, 9, \text{“high”}\}$. As you can see, the integers greater than nine have been grouped into a bin called "high." One can even use integers regardless of sign, with a mapping like $\{\text{“low”}, -2, -1, 0, 1, 2, \text{“high”}\}$.

**A Realistic Example** Suppose several million newspaper articles have been digitized, and it is desired to map them into some basic classes: sports, business, crime, politics, fashion, or other. When classifying newspaper articles, there could be $n$ words that are searched for. The values of the $X_i$s could be boolean, to represent the presence or absence of each word in some article. Alternatively, they could be non-negative integers, to represent the counts of the words. Perhaps the values of $X_i = \{0, 1, 2\}$ represent the word occurring not at all (0), only once (1), or multiple times (2).

**The "Training" Process** Like most machine learning algorithms, there is a learning stage and an operating stage. For each attribute (random variable) $X_k$, and each class $c_j$, it is necessary to know $Pr[(X_k = x_k)|(C = c_j)]$ which is the probability that the $k$th attribute ($X_k$) will take any particular value $x_k$, given that the object is from class $c_j$. The pre-computation of these conditional probabilities is called "training" the classifier. Typically, this is done by having a "training set" of many objects, each of which has been marked carefully with its correct classification (determined manually or by some other method), before the training starts. Once the training is complete, the Bayesian Classifier is ready to examine new objects (not seen before) and guess their classification.

In the previous example, perhaps an intern is hired to manually classify 5000 articles, that will comprise the training set. Then the several million articles can be classified by the Bayesian Classifier.

**Continuous Random Variables** For a continuous random variable, such as human heights, the possible values should be collected into bins. For example, each bin might represent a range of heights 2 inches or 5 cm apart, with one additional bin for persons shorter than some cut-off height (e.g. 59 inches or 150 cm) and one additional bin for persons taller than some cut-off height (e.g. 79 inches or 201 cm). According to this strategy, human heights would be allocated to 12 bins.

**The Classification Process** Whenever a new object arrives for classification, its particular attributes for each of the random variables comprising $\vec{X} = (X_1, X_2, X_3, \ldots, X_n)$ are measured and recorded as $\vec{x} = (x_1, x_2, x_3, \ldots, x_n)$. Once this is done, a probability for each class $c_i$ is computed with the following formula

$$Pr[(C = c_i)|(\vec{X} = \vec{x})] = \frac{Pr[C = c_i] \prod_{k=1}^{k=n} Pr[(X_k = x_k)|(C = c_i)]}{\sum_{j=1}^{j=m} Pr[C = c_j] \prod_{k=1}^{k=n} Pr[(X_k = x_k)|(C = c_j)]}$$

or alternatively, a "score" for each class is computed by

$$s_i = Pr[C = c_i] \prod_{k=1}^{k=n} Pr[(X_k = x_k)|(C = c_i)]$$

Once all those probabilities (or scores) have been computed, the classifier will output a guess corresponding to whichever class achieved the highest probability (or score). By the way, a few published papers have those two formulas incorrectly, a point explored further in Appendix B.3.

## 4.2 Can we Circumvent these Independence Assumptions?

The derivation of the Bayesian Classifier Formula (done in detail in Appendix B) makes the following $(n - 1)$ assumptions, for each of the $m$ classes:

$$\begin{aligned}
Pr[(X_2 = x_2)|((C = c_i) \cap (X_1 = x_1))] &= Pr[(X_2 = x_2)|(C = c_i)] \\
Pr[(X_3 = x_3)|((C = c_i) \cap (X_1 = x_1) \cap (X_2 = x_2))] &= Pr[(X_3 = x_3)|(C = c_i)] \\
Pr[(X_4 = x_4)|((C = c_i) \cap (X_1 = x_1) \cap (X_2 = x_2) \cap (X_3 = x_3))] &= Pr[(X_4 = x_4)|(C = c_i)] \\
\vdots &= \vdots \\
Pr[(X_n = x_n)|((C = c_i) \cap (X_1 = x_1) \cap (X_2 = x_2) \cap \cdots \cap (X_{n-1} = x_{n-1}))] &= Pr[(X_n = x_n)|(C = c_i)]
\end{aligned}$$

The reader is probably surprised to see those $(n - 1)$ assumptions for each of $m$ classes, implying $(n - 1)m$ assumptions in total. These assumptions are so specific and bizarre that Bayesian Classifiers are sometimes called

"Naïve Bayesian Classifiers." By the way, an equivalent formulation of those assumptions is given in Appendix B.4. Note well that with $n = 1$, there are no assumptions. The absence of assumptions reinforces the view that Bayes' Rule is on extremely solid ground, in comparison to the spam formula and Bayesian Classifiers.

This paper would like to point out that the probabilities

$$Pr[(X_1 = x_1, X_2 = x_2, X_3 = x_3, \ldots, X_n = x_n)|(C = c_j)]$$

can be computed by experimentation, in small cases—a point we will make precise momentarily. This would eliminate the need for the assumptions, and it would make the computations of the probabilities $Pr[(C = c_i)|(\vec{X} = \vec{x})]$ exact.

If all the $X_i$s were boolean, then there are $2^n$ possible values of $\vec{x}$. This means that there are $m2^n$ possible probabilities of the form $Pr[(\vec{X} = \vec{x})|(C = c_j)]$ that need to be computed by experiment. If using $n = 6$ boolean attributes to classify objects into, for example, 8 classes, then that's only $8(2^6) = 8(64) = 512$ experiments. This is not at all a computationally difficult task. Of course, if using $n = 20$ boolean attributes to classify objects into 64 classes, then that's $64(2^{20}) = 2^{26} = 67,108,864$ experiments, which might be inconvenient or require far too much training data.

More generally, suppose that the $X_i$s are chosen from finite sets. In Section 4.1, we saw how human height, a continuous random variable, could be mapped to 12 bins. In similar manner, we can accommodate other continuous or discrete-but-infinite random variables into bins to make those random variables into finite sets. Let the cardinality of the set of possible choices for $X_i$ be denoted $S_i$. Then the number of experiments required is given by $m(S_1)(S_2)(S_3)\cdots(S_n)$ which might not be too large in many cases.

In summary, the independence assumptions are not needed, and experimentation could provide for exact probabilities, instead of approximate probabilities. In some cases, too many experiments would be required. However, in other cases, the number of experiments required is modest and entirely feasible.

### 4.3 The Assumption of Equiprobable Classes in Bayesian Classifiers

In online forums and blog posts, one tends to see an alarming error. Of the $m$ classes $C_1, C_2, C_3, \ldots, C_m$, for some reason the assumption that $Pr[C = C_i] = 1/m$ is made for each possible $C_i$. This error was only found in 3 out of 25 published research papers using Bayesian reasoning in cybersecurity, so a discussion of it has been moved to Appendix B.3 and Appendix G.

## 5 Acknowledgements

The author is deeply indebted to Prof. Jing Xi of the University of Wisconsin—Stout's Department of Mathematics, Statistics, and Computer Science, for assistance with conditional independence and with the continuous forms of Bayes' Rule presented in Appendix A. The undergraduate, Kyle Conway, of the Cybersecurity concentration of the Applied Mathematics & Computer Science program at the University of Wisconsin—Stout was extremely helpful with the literature search in Appendix H and with the preparation of the bibliography of this paper. Thanks must be given to the Vice-Dean Dr. Marianna Durcheva, of the Technical University of Sophia, who encouraged this author to develop, finalize, and publish this paper, at a time when it was nothing more than a collection of slides. The organizers of *The 43rd International Conference on the Applications of Mathematics in Engineering and Economics*, in Sozopol, Bulgaria, were very kind in giving that talk the "best presentation" award, for which this author is very grateful.

## REFERENCES

[1] Altwaijry, H., and Algarny, S. Original article: Bayesian based intrusion detection system. *J. King Saud Univ. Comput. Inf. Sci. 24*, 1 (Jan. 2012), 1–6.

[2] Amor, N. B., Benferhat, S., and Elouedi, Z. Naive bayes vs decision trees in intrusion detection systems. In *Proceedings of the 2004 ACM Symposium on Applied Computing* (New York, NY, USA, 2004), SAC '04, ACM, pp. 420–424.

[3] Axelsson, S. The base-rate fallacy and the difficulty of intrusion detection. *ACM Transactions on Information and System Security (TISSEC) 3*, 3 (2000), 186–205.

[4] Burroughs, D. J., Wilson, L. F., and Cybenko, G. V. Analysis of distributed intrusion detection systems using bayesian methods. In *Proceedings of IEEE International Performance Computing and Communication Conference* (2002), pp. 329–334.

[5] Farid, D. M., Harbi, N., and Rahman, M. Z. Combining naive bayes and decision tree for adaptive intrusion detection. *International Journal of Network Security and Its Applications 2*, 2 (Apr. 2010), 12–25.

[6] Firdausi, I., Erwin, A., Nugroho, A. S., and Lim, C. Analysis of machine learning techniques used in behavior-based malware detection. In *Advances in Computing, Control and Telecommunication Technologies (ACT), 2010 Second International Conference on* (2010), IEEE, pp. 201–203.

[7] Kim, H., and Singh, C. Security analysis for system operation using Bayes classifier. In *Power Engineering Society General Meeting, 2003, IEEE* (2003), vol. 2, IEEE, pp. 661–666.

[8] Koc, L., Mazzuchi, T., and Sarkani, S. A network intrusion detection system based on a hidden naïve bayes multiclass classifier. *Expert Systems with Applications 39*, 18 (2012), 13492–13500.

[9] Kolter, J. Z., and Maloof, M. A. Learning to detect malicious executables in the wild. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (New York, NY, USA, 2004), KDD '04, ACM, pp. 470–478.

[10] Kolter, J. Z., and Maloof, M. A. Learning to detect and classify malicious executables in the wild. *Journal of Machine Learning Research 7* (2006), 2006.

[11] Ma, Y., Liang, S., Chen, X., and Jia, C. The approach to detect abnormal access behavior based on naive bayes algorithm. In *The Proceedings of the 2016 10th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS'16)* (2016), IEEE, pp. 313–315.

[12] Mir, N. M., Khan, S., Butt, M. A., and Zaman, M. An experimental evaluation of bayesian classifiers applied to intrusion detection. *Indian Journal of Science and Technology 9*, 12 (2016).

[13] Mukherjee, S., and Sharma, N. Intrusion detection using naive bayes classifier with feature reduction. *Procedia Technology 4* (2012), 119–128.

[14] Panda, M., and Patra, M. R. Network intrusion detection using nave bayes. *International Journal of Computer Science and Network Security 7*, 12 (Dec. 2007).

[15] Raphel, J., and Vinod, P. Heterogeneous opcode space for metamorphic malware detection. *Arabian Journal for Science and Engineering 42*, 2 (2017), 537–558.

[16] Rosen, K. *Discrete Mathematics and Its Applications*, 7th ed. McGraw Hill, 2012.

[17] Santos, I., Brezo, F., Ugarte-pedrero, X., and Bringas, P. G. Opcode sequences as representation of executables for data-mining-based unknown malware detection, information sciences 227. *Information Sciences 231* (2013), 64–82.

[18] Sariman, G., and Kucuksille, E. U. A novel approach to determine software security level using bayes classifier via static code metrics. *Elektronika ir Elektrotechnika 22*, 2 (2016), 73–80.

[19] Sayfullina, L., Eirola, E., Komashinsky, D., Palumbo, P., Miche, Y., Lendasse, A., and Karhunen, J. Improved naive bayes classifier for android malware classification. In *The Proceedings of the 14th IEEE International Conference on Trust, Security and Privacy in Computing and Communications (IEEE TrustCom'15)* (Aug. 2015), IEEE.

[20] Schultz, M. G., Eskin, E., Zadok, E., and Stolfo, S. J. Data mining methods for detection of new malicious executables. In *Proceedings of the IEEE Symposium on Security and Privacy* (2001), pp. 38–49.

[21] Sebyala, A. A., Olukemi, T., and Sacks, L. Active platform security through intrusion detection using naive bayesian network for anomaly detection. In *Proceedings of London communications symposium* (2002).

[22] Tsai, C.-F., Hsu, Y.-F., Lin, C.-Y., and Lin, W.-Y. Intrusion detection by machine learning: A review. *Expert Syst. Appl. 36*, 10 (2009), 11994–12000.

[23] Viegas, E., Santin, A. O., França, A., Jasinski, R., Pedroni, V. A., and Oliveira, L. S. Towards an energy-efficient anomaly-based intrusion detection engine for embedded systems. *IEEE Transactions on Computers 66*, 1 (2017), 163–177.

[24] Villamarín-Salomón, R., and Brustoloni, J. C. Bayesian bot detection based on dns traffic similarity. In *Proceedings of the 2009 ACM symposium on Applied Computing* (2009), ACM, pp. 2035–2041.

[25] Xiang, C., Yong, P. C., and Meng, L. S. Design of multiple-level hybrid classifier for intrusion detection system using bayesian clustering and decision trees. *Pattern Recognition Letters 29*, 7 (2008), 918–924.

[26] Yerima, S. Y., Sezer, S., McWilliams, G., and Muttik, I. A new android malware detection approach using bayesian classification. In *Advanced Information Networking and Applications (AINA), 2013 IEEE 27th International Conference on* (2013), IEEE, pp. 121–128.