Modeling Learningless Vulnerability Discovery using a Folded Distribution

Awad A. Younis[1], **HyunChul Joh**[1], **and Yashwant K. Malaiya**[1]
[1]Computer Science Department, Colorado State University, Fort Collins, CO 80523, USA

Abstract – *A vulnerability discovery model describes the vulnerability discovery rate in a software system, and predicts the future behavior. It can allow the IT managers and developers to allocate their resources optimally by timely development and application of patches. Such models also allow the end-users to assess security risk in their systems. Recently, researchers have proposed a few vulnerability discovery models. The models are based on different assumptions, and thus differ in their accuracy and prediction capabilities. Among these models, the AML model has been found to have performed better in many cases in terms of model fitting and prediction capabilities. The AML model assumes that the discovery rate is symmetric. However, it has been noted that there are cases when the discovery trend is asymmetric. In this paper, we investigate the applicability of using a new vulnerability discovery model called Folded model, based on the Folded normal distribution, and compare it with the AML model. Results show that Folded model performs better than the AML model in general for both model fitting and prediction capabilities in cases* when the learning phase is not present.

Keywords – Software security; vulnerability discovery model (VDM); Folded model; Risk assessment

1 Introduction

The society today relies on the Internet not only for activities such as sending emails, searching the net, and reading news but also security critical tasks such as checking bank account, and online purchasing. As a result, security has become the main concern for both vendors and users of services. Not only the network and communication infrastructure but also software systems themselves at end-nodes need to be secured. Having vulnerabilities, which are software defects that might be exploited by a malicious user causing loss or harm [1], in such systems, could potentially cause a lot of damage. The Code Red worm [2], a computer worm that exploited vulnerabilities existing in Microsoft's IIS (Internet Information Services) in 2001, is an example of the damage that can occur due to presence of vulnerabilities.

Evaluating security quantitatively in software systems is required to achieve an optimal security level. A few quantitative vulnerability discovery models (VDMs) have recently been proposed. They include Rescorla's exponential model [3], Anderson's thermodynamic model [4], and Alhazmi-Malaiya Logistic (AML) model [5], each of them is based on its own assumptions and is characterized by its specific parameters. The VDMs let developers project the future behavior of the vulnerability discovery processes. The VDMs are essential for two main reasons. First, they allow developers to optimally allocate their resources that will be needed for developing patches for the security holes quickly. Also, they allow the users to assess the potential risk due to new vulnerabilities.

Investigating the prediction capability and accuracy of these models has been studied by Alhazmi and Malaiya [6]. It has been found that the AML model generally fits the data for several software systems better than other models. The AML model is obtained using the assumption that as the market share of a software increases, the rate of vulnerability discovery also increases. When the software starts losing its market share, or when there are a few vulnerabilities remaining to be found, the vulnerability discovery rate decreases [5]. Thus, the motivation of the vulnerability finders, both white hat and black hat, is driven by the market share. The AML model is logistic, and thus the increase and decrease in the discovery process is assumed to be symmetric around the peak. However, it has been noted [7][8], that the discovery rate may not be necessarily symmetrical. This limitation of the AML model can possibly be addressed using alternative models that capture asymmetric behavior.

Kim [7], and Joh and Malaiya [9] have shown that asymmetric VDMs are feasible and have better performance than the symmetric models in some cases. In this paper, we examine the Folded model suggested by Kim [7], as an alternative VDM. Kim however did not examine the model using actual datasets. Here, we examined the applicability of the Folded VDM using actual vulnerability discovery data for four popular software systems. Specifically, we compare the Folded and AML models using goodness of fit and prediction capabilities for these datasets.

The paper is organized as follows. Section 2 presents the background and the related literature on the VDMs. In Section 3, the AML model is discussed and its potential limitations are identified. In section 4, the Folded VDM will be introduced. Section 5 presents the results of the comparison of the AML and Folded models using goodness of fit tests and prediction capabilities. Finally, the concluding comments are given along with the issues that need further research.

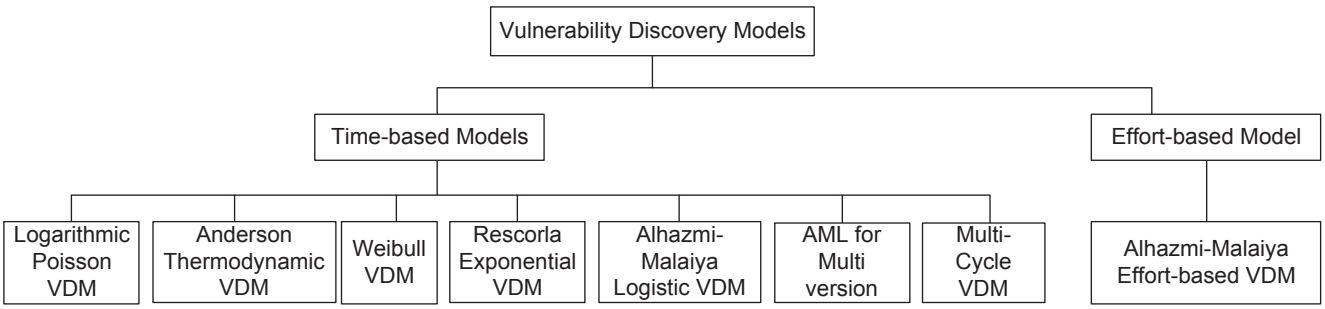

Figure 1. Taxonomy for Vulnerability Discovery Models

2 The Vulnerability Discovery Models

The VDMs proposed recently are somewhat analogous to software reliability growth model (SRGM), but there are significant differences. VDMs are probabilistic models for modeling the discovery rate of vulnerabilities in software systems [10]. These models use the historical data such as release date, the discovery date of vulnerabilities and possibly the system usage data. While the vulnerabilities are security related defects, they tend to be treated differently compared with ordinary software defects [11][12]. Normal defects found after release are frequently ignored and not fixed until the next release because they do not represent a high degree of risk. On the other hand, software developers need to patch vulnerabilities right after they are found, due to the high risks they represent. The security issues can greatly impact not only organizations such as banks, brokerage houses, on-line merchants, government offices but also individuals.

Quantitative risk analysis of systems with a continual vulnerability discovery has only recently started to be investigated. A few VDMs proposed by researchers include Anderson [4], Rescorla [3], Kim [7], Alhazmi and Malaiya Logistic model [13], Alhazmi and Malaiya Effort based model [14], Ozment and Schechter [15], and Chen et al. [16]. Figure 1 shows classification of vulnerability discovery models. Each model has its own mathematical representation and parameters. As a result, different VDMs can make somewhat different projections using the same data. No specific guidance is currently available about which models should be used in a given situation.

Rescorla [3] has introduced quadratic and exponential VDMs. He fitted the proposed models but did not evaluate their predictive accuracy. Anderson [4] proposed a thermodynamic vulnerability discovery model, but did not apply the model to any actual data. Alhazmi and Malaiya [5] proposed the logistic vulnerability discovery model, termed the AML model. The AML model presumes a symmetric software vulnerability discovery process. This model has shown a good statistically significant goodness-of-fit for the well-known operating systems such as Windows and Red Hat Linux, and some Internet applications such as browsers and HTTP servers. Its predictive capability was tested by Alhazmi and Malaiya [6] and it has shown good results. In another study [13], they found that the AML model provides a better goodness-of-fit compared to Rescorla and Anderson models.

Alhazmi and Malaiya [14] have also proposed an effort-based model which utilizes the number of system installations as the independent factor instead of calendar time. They argued that it is much more rewarding to discover a vulnerability in a system which is installed on a large number of computers. However, the effort-based model requires the number of users for a target product in market share which is not always easy to be obtained. Woo et al. [2] have examined the goodness-of-fit as well as the prediction capability for the effort-based model.

Joh et al. [8] have studied Weibull VDM, which was first proposed by Kim [7]. They argued that the assumption made by the AML model that the rate of discovering vulnerability is symmetric around the peak value is not always true. They used Weibull distribution to capture the asymmetric behavior as an alternative to the AML model. However, the Weibull model did not always provide a good fit.

3 The Symmetrical AML VDM

The AML VDM [1] is a time-based model. It assumes that at the release of the software the vulnerability discovery rate increases gradually. This is known as the learning phase in which the software gains market share and installed bases remain small. After the learning phase, the system starts to attract more users and the number of vulnerabilities grows linearly. In this phase, which is known as the linear phase, the maximum vulnerability discovery rate is obtained by finding the slope. The learning phase is considered as the most important phase because most of the vulnerabilities will be discovered during this phase. However, when the system starts to be replaced by a newer version and users start to switch to the next version and as a result the vulnerability finders start to lose interest in finding vulnerabilities in the older version. As a result, the vulnerability discovery rate drops. Therefore, the cumulative number of vulnerabili-

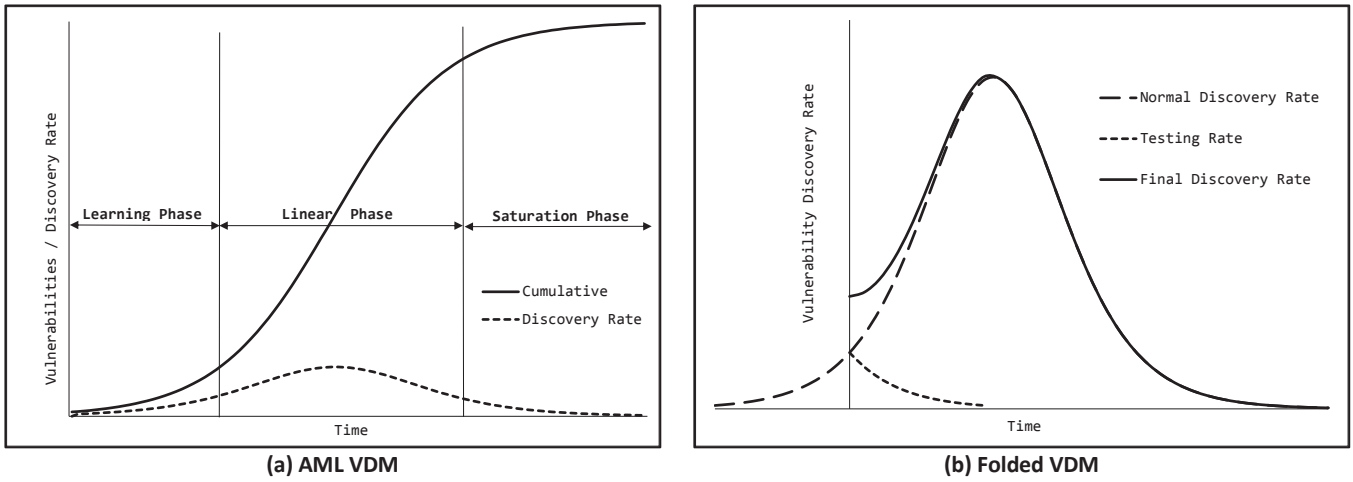

Figure 2. Vulnerability discovery process and rates for AML and Folded VDMs

ties becomes stable. The three phases are shown in Figure 2 (a).

The AML model assumes that the vulnerability discovery processes are controlled by the market share of the software and the number of the undiscovered vulnerabilities. The model assumes that the vulnerability discovery rate is given by the differential equation:

$$\frac{d\Omega}{dt} = A\Omega(B - \Omega) \quad (1)$$

Equation (1) has two factors. The first factor $A\Omega$, where A is a constant, increases as the market share increases, and $(B - \Omega)$, where B represents the total number of vulnerabilities, decreases as the remaining vulnerabilities decreases. Equation (1) can be solved to obtain the logistic expression for $\Omega(t)$:

$$\Omega(t) = \frac{B}{BCe^{-ABt}+1} \quad (2)$$

Note that $\Omega(t)$ aproaches B as the calendar time t approaches infinity. The parameters A and C determine the shape of the curve [13]. C is a constant introduced while solving Equation (1).

AML model assumes a symmetrical vulnerability discovery rate as shown by the dotted curve in Figure 2 (a). Although the AML model has been found to fit real data of many software systems, there is no compelling reason why the rise and fall should be symmetric since they may be controlled by different factors. Some datasets do show a noticeable asymmetry [9]. These findings violate the symmetric assumption made by this model. Thus, looking for alternative VDMs that can deal with this trend is needed.

Actual data can show a departure from the s-shape assumed by the logistic model. In many cases, a software system gradually evolves as code is modified or patched or additional code is added. This will inject new vulnerabilities into the system which will delay the onset of saturation. In many cases, a new version is widely anticipated and is adapted by many users soon after its release. This will result in the learning period to shrink or even disappear. In the next section we consider the Folded VDM that offers the capability of modeling the behavior when the learning period is very small.

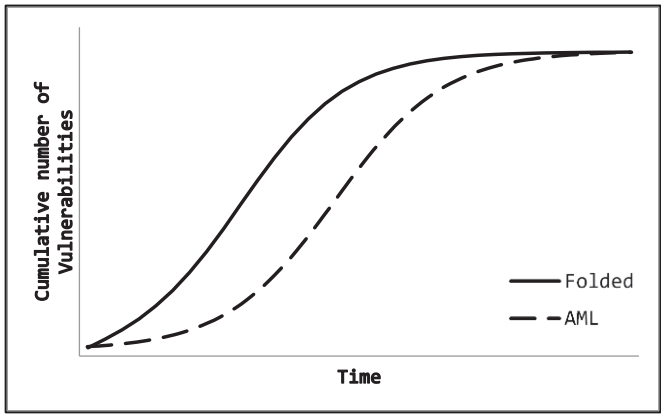

Figure 3. General cumulative vulnerability discovery trends

4 Asymmetrical Folded VDM

The normal distribution is symmetric around its mean and is defined for a random variable that takes values from $-\infty$ to $+\infty$. In some cases, a distribution is needed that has no negative values. Daniel [17] had proposed a half-normal distribution that folds the normal distribution at the mean that now corresponds to value zero. A more general version of it was proposed by Leone et al. [18] which is termed a Folded normal distribution that is defined for a random variable taking values between 0 and $+\infty$. It is obtained by folding the negative values into the positive side of the distribution. Whenever measurements of a normally distributed random variable are taken and the algebraic sign is discarded, the resulting distribution will be a Folded distribution. The folded distribution has been found usable in industrial practices such as measurement of flatness, straightens, and determination of the centrality of the sprocket holes in motion picture film [18]. The probability density function (pdf)

and the cumulative distribution function (cdf) of the distribution are both derived from their counterparts in the normal distribution, pdf and cdf.

The Folded distribution, as applied to vulnerability discovery, is illustrated in Figure 2 (b). The vulnerability discovery starts at time $t = 0$ which corresponds to the release time of the software. Since the initial value is non-zero because of the contribution of folding, the learning period is minimized as shown in Figure 3. Hence, here, we propose the Folded VDM as an asymmetrical model as suggested by Kim [7]. The proposed vulnerability discovery rate of the Folded model is given by Equation (3).

$$f(t) = \frac{\gamma}{\sqrt{2\pi}\,\sigma}\left[e^{-\left(\frac{(t-\tau)^2}{2\sigma^2}\right)} + e^{-\left(\frac{(t+\tau)^2}{2\sigma^2}\right)}\right], t \geq 0 \quad (3)$$

Here, t represents the calendar time, τ is a location parameter, σ is a scale parameter, and γ represents the number of vulnerabilities that will be eventually discovered. The second term in Equation (3) represents the part of the distribution folded to the positive side as shown in Figure 2 (b) which shows the discovery process for the Folded VDM. The cumulative number of vulnerabilities described by Folded VDM is presented in Equation (4).

$$F(t) = \frac{\gamma}{2}\left[erf\left(\frac{(t-\tau)}{\sqrt{2\sigma}}\right) + erf\left(\frac{(t+\tau)}{\sqrt{2\sigma}}\right)\right], t \geq 0 \quad (4)$$

where $erf(\cdot)$ is the error function which is used to calculate the integral from zero. Figure 3 shows the cumulative Folded vulnerability discovery process along with the behavior of AML. Figure 3 also shows the lack of the learning phase for the Folded model.

Compared to AML, the Folded VDM has shorter learning phase or missing learning phase which makes the normal distribution asymmetric. It results in a higher discovery rate at the beginning which may be especially applicable to the cases where $\Omega(t)$ plot is linear even at the beginning.

5 Model comparisons and observations

We have fitted the AML and Folded VDMs to the four datasets: Windows 7, OSX 5.x, Apache Web Server 2.0.x, and Internet Explorer 8. Table 1 shows released dates, market shares and the number of vulnerabilities in each system. These software systems have been chosen because they have relatively short learning phase, and thus they can be used to test whether the proposed Folded model is capable of capturing the learningless vulnerability discovery trend. Figure 4 shows model fittings for the two VDMs on the four datasets. While visually both models appear to fit well, in the next section we analyze the goodness of fit by evaluating the p-values.

5.1 Goodness of Fit analysis

Table 2 shows the model parameters along with the p-values of χ^2 goodness of fit tests. The χ^2 statistic (χ_s^2) is calculated as:

Table 1. Datasets Used

	Released	*Vuln.	Share(%)
Win 7	2009-JUL	80	**25.11
OSX5.x	2007-OCT	211	**1.30
Apache 2.0.x	2000-MAR	68	***62.71
IE 8	2009-MAR	72	**33.06

*http://nvd.nist.gov/ on JAN 2011. Only after the released date.
**http://marketshare.hitslink.com/ on APR 2011.
***http://news.netcraft.com/ on May 2011. For total version.

$$\chi_s^2 = \sum_{i=1}^{n} \frac{(o_i - e_i)^2}{e_i} \quad (5)$$

where o_i and e_i are the observed and expected values at i^{th} time point respectively. The null hypothesis for the test is that the actual distribution is well described by model fittings. Hence, in Table 2, p-value close to 1 means good model fitting whereas less than 0.05 is considered as not being statistically significant when we select the α level as 0.05.

Figure 4 suggests that all the datasets show linear discovery trends for the period examined and either do not have a learning phase or it is very short. The main reason for linearity in the early part can be because of quick adoption of the version considered as a result of the anticipation of the release. Both the users and the vulnerability finders are not waiting for the software to become sufficiently popular, they take it for granted that it will be. During the later part, the linear behavior could be that since the systems are continually evolving, new code is being injected time to time which introduces additional vulnerabilities. The saturation phases would not be seen in the vulnerability discovery process for such systems until they stop evolving. In general, we observed that Folded VDM captures the starting and ending data points better than AML model for these datasets.

P-values in Table 2 indicate that all the model fittings are statistically significant since p-value is greater than 0.05. Windows 7 and Internet Explorer 8 fit the Folded model better whereas AML fits OSX 5.x slightly better. Apache 2.0.x data fits both models very well with p-value 1. However, visual inspection tells that Folded model performs better at the beginning and the end of the time period. Folded model provides p-values which are consistently greater than 0.9 while AML has a lower value in the two cases.

5.2 Prediction capabilites

The main use of a model is predicting the future trends based on the available data, rather than reviewing the past behavior. In that sense, prediction capability should be considered more important than model fitting. Models having good fitting results may not necessarily possess good prediction abilities of the process behavior changes with time.

We use two normalized prediction capability measures [19], Average Error (AE) and Average Bias (AB), as given in Equation (6) and (7) respectively. AE is a measure of how well a model predicts throughout the time period, and AB indicates the general bias of the model which assesses its tendency to overestimate or underestimate.

$$AE = \frac{1}{n}\sum_{t=1}^{n}\left|\frac{\Omega_t - \Omega}{\Omega}\right| \quad (6)$$

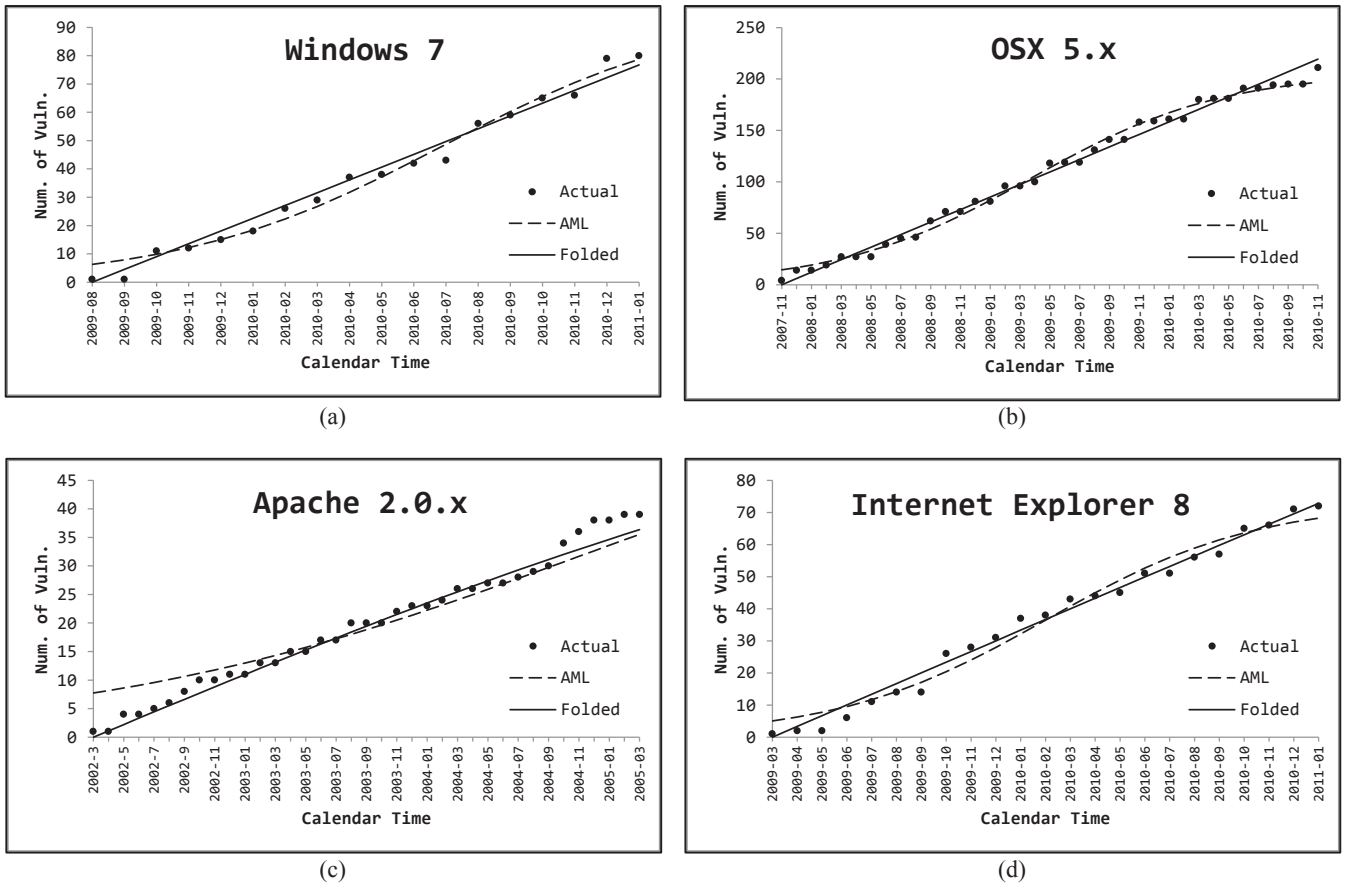

Figure 4. Model fitting for AML and Folded VDMs

Table 2. χ^2 Goodness of fit tests

	AML				Folded			
	A	B	C	P-value	τ	σ	γ	P-value
Win 7	2.52E-03	96.75671	0.148963	0.6970	0.063742	11381.63	64427.18	0.9673
OSX5.x	7.49E-04	206.8057	0.064464	0.9845	0.063742	1969.209	15029.21	0.9428
Apache 2.0.x	9.88E-04	62.69023	0.113327	1.0000	0.065227	47.81145	66.26354	1.0000
IE 8	3.22E-03	73.39949	0.185473	0.7337	0.065227	97.30989	407.1494	0.9839

$$AB = \frac{1}{n}\sum_{t=1}^{n}\frac{\Omega_t - \Omega}{\Omega} \quad (7)$$

In the equations, n is a total number of time points (in months in this case), and Ω is the actual number of total vulnerabilities. Ω_t is the estimated number of total vulnerabilities at time t. The normalized prediction error values for each time point are plotted in Figure 5. The x-axis represents the time as a percentage where 0% and 100% correspond to the release date and the final data point that the model is attempting to predict. Table 3 shows the values for AE and AB.

The error plots in Figure 5 show that the Folded model provides a more stable prediction with a significantly less error in most situations. In Table 3, the AB and AE values show that the Folded model almost always performs better than AML. For Windows 7, OSX 5.x and Internet Explorer 8, Folded model outperformed the AML. For Apache 2.0.x, the two models result in somewhat similar outcomes for the AE value.

6 Conclusion & Future work

This paper examines a new vulnerability discovery model based on the folded normal distribution and evaluates its applicability using real datasets for four major software products. It also compares the new proposed model with the symmetrical AML vulnerability discovery model.

Software developers need to estimate the resources needed for development of patches for the vulnerabilities that are likely to be found in future. A quick patch release after the discovery of a vulnerability will significantly reduce the security risk to the organizational and individual users. An organization needs to assess the resources needed to address

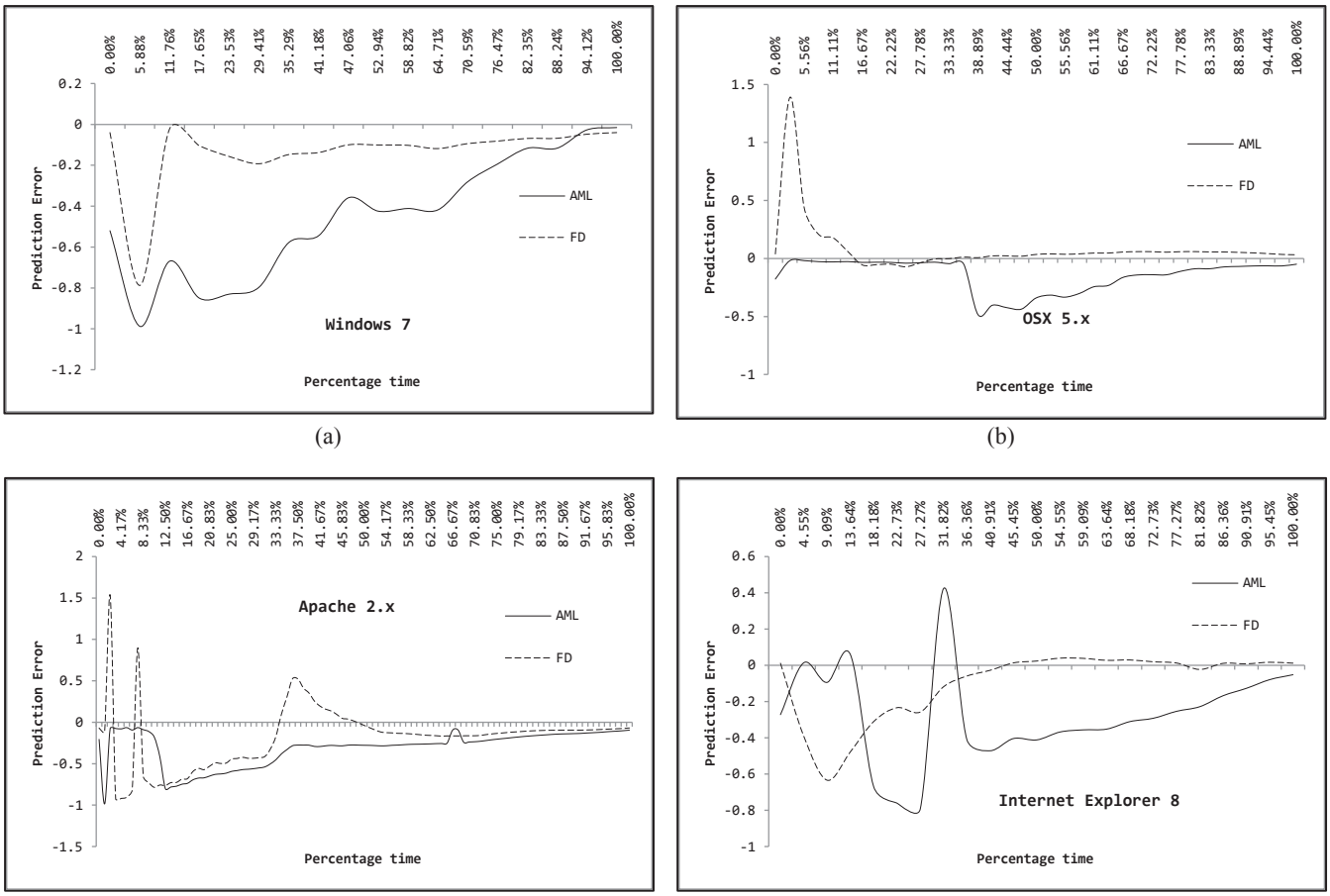

Figure 5. Prediction errors for AML and Folded VDMs

Table 3. Average Bias & Average Error (%-Time: 0% ~ 100%)

	AB		AE	
	AML	Folded	AML	Folded
Win 7	-0.45222	-0.13405	0.452221	0.134048
OSX5.x	-0.14514	0.081817	0.145141	0.096575
Apache 2.0.x	-29.6239	-17.7062	29.62394	28.87495
IE 8	-0.27722	-0.09876	0.320391	0.121494

future vulnerabilities; including the patch application effort and reserve resources needed to alleviate the impact of possible intrusions. Both of these require the use of a vulnerability discovery model that can make sufficiently accurate vulnerability discovery rate projections.

The AML model is the only model that has been formulated to specifically describe the discovery process. The fitting and prediction capability of the AML model has been found to be better than other models for most datasets. However, it has also been found that the discovery trends can be different in different circumstances. In one hand, for the software systems that has been in the market for long period of time, their behavior have been found to be better described by symmetric models such as AML logistic model which exhibits both learning and a saturation phases in addition to the linear phase. On the other hand, some systems have a vulnerability discovery rate that tends to be linear from the beginning and thus lack a learning phase.

In this paper we have formally defined and investigated the Folded vulnerability discovery model based on folded normal distribution which is asymmetric by definition and can represent a learningless discovery process. Its model fitting and prediction capabilities have been tested and compared with the AML model for four popular software systems. While both Folded and AML models have been found to fit the vulnerabilities datasets of Windows 7, OSX 5.x, Apache Web server 2.0.x and Internet Explorer 8 well, they differ significantly in the prediction capability. The short learning phase is apparently captured by the Folded model much better than the AML logistic model for the four da-

tasets. The folded model consistently outperforms the AML model in terms of the prediction capabilities for the datasets with no learning phase.

The Folded model needs to be further investigated by applying it to as many software systems as possible and comparing it with other competing models. That will allow development of guidelines as to when this model would be most suitable. The significance of the parameters also needs to be examined.

7 References

[1] C. P. Pfleeger and S. L Pfleeger. *Security in Computing*, 3rd ed. Prentice Hall PTR, 2003.

[2] S. Woo, H. Joh. O. Alhzmi, and Y. Malaiya. Modeling vulnerability discovery process in Apache and IIS HTTP servers, *Computers &Security*, 30(1), 2011, pp. 50-62.

[3] E. Rescorla. Is finding security holes a good idea? *IEEE Security & Privacy*, 3(1), 2005, pp.14-19.

[4] R. Anderson. Security in open versus closed systems - the dance of boltzmann, coase and moore. Proc. Conf. on Open Source Software: Economics, Law and Policy, 2002, pp. 1–15.

[5] O. Alhazmi, Y. Malaiya, and I. Ray. Security vulnerabilities in software systems: a quantitative perspective. Proc. IFIP WG 11.3 Working Conference on Data and Applications Security, 2005, pp. 281–294.

[6] O. Alhazmi and Y. Malaiya. Measuring and enhancing prediction capabilities of vulnerability discovery models for apache and IIS http servers. Proc. Int. Symp. Software Reliability Eng. (ISSRE), November 2006, pp. 343–352.

[7] J. Kim. Vulnerability discovery in multiple version software systems: open source and commercial software systems, Master thesis, Colorado State University, 2007.

[8] H. Joh, J. Kim, and Y. Malaiya. Vulnerability discovery modeling using Weibull distribution. Proc. Int. Symp. Software Reliability Eng. (ISSRE), November 2008, pp. 343–352.

[9] H. Joh and Y. K. Malaiya. Modeling skewness in data with S-shaped vulnerability discovery models, Proc. Int. Symp. Software Reliability Eng. (ISSRE), November 2010, pp. 406-407.

[10] A. Ozment. *Vulnerability Discovery & Software Security*. PhD dissertation, University of Cambridge. August 31, 2007.

[11] P. Anbalagan and M. Vouk. On reliability analysis of open source software - fedora. Proc. Int. Symp. Software Reliability Eng. (ISSRE), November 2008, pp. 325–326.

[12] T. Zimmermann, N. Nagappan, and L. Williams. Searching for a needle in a haystack: predicting security vulnerabilities for windows vista. Proc. Int. Conf. on Software Testing, Verification and Validation. (ICST), 2010 , pp. 421–428.

[13] O. Alhazmi and Y. Malaiya. Modeling the vulnerability discovery process. Proc. Int. Symp. Software Reliability Eng. (ISSRE), November 2005, pp.129–138.

[14] O. Alhazmi and Y. Malaiya. Quantitative vulnerability assessment of system software, Proc. Ann. IEEE Reliability and Maintainability Symp. 2005, pp. 615-620.

[15] A. Ozment and S. Schechter. Milk or wine: does software security improve with age? Proc. 15th Usenix Security Symposium, Vancouver, Canada, 2006, pp. 93-104.

[16] K. Chan, D-G Feng, P-R Su, C-J Nie, and X-F Zhang. Multi-cycle vulnerability discovery model for prediction. *Journal of Software*, 21(9), 2010, pp. 2367-2375.

[17] C. Daniel. Use of Half-normal plots in interpreting factorial two-level experiments, *Technometrics* , 1(4), 1959. pp. 311-341.

[18] F. C. Leone, L. S. Nelson, and R. B. Nottingham. The folded normal distribution. *Technometrics*, 3(4), 1961, pp. 543-550.

[19] Y. K. Malaiya, N. Karunanithi, and P. Verma. Predictability of software reliability models. *IEEE Transactions on Reliability*, 41(4), 1992, pp. 539-546.