Android Güvenlik Açıklarının Modellenmesi: İstatistiksel Dağılımlardan Analizler[1]

***

Modeling Android Security Vulnerabilities: Insights from Statistical Distributions

Kerem GENCER[2]
Fatih BASCİFTCİ[3]

DOI:10.33461/uybisbbd.1524207

Makale Bilgileri

Makale Türü: Araştırma Makalesi

Geliş Tarihi: 31.07.2024

Kabul Tarihi: 05.09.2024

Öz

Android işletim sistemi, multimedya özelliklerini destekleyen bir mobil işletim sistemidir. Android, ses, video, resim ve diğer multimedya içeriklerini oynatmak, kaydetmek, düzenlemek ve paylaşmak için çok çeşitli uygulamalar ve entegre özellikler sunar. Çoğu Android cihazda kamera, hoparlör, mikrofon ve diğer multimedya bileşenleri bulunur. Yazılım güvenliğinde, güvenlik açıkları genellikle yazılım geliştirme sırasında ortaya çıkan kritik endişelerdir. Bu güvenlik açıklarını sürümden sonra tahmin etmek, risk değerlendirmesi ve azaltma için önemlidir. Çeşitli modeller araştırılmış olsa da Android işletim sistemi nispeten keşfedilmemiş durumdadır. Bu çalışma, yaygın olarak kullanılan Alhazmi-Malaiya Lojistik (AML) modeline uygunluklarını karşılaştırarak, farklı istatistiksel dağılımlar kullanarak Android güvenlik açıklarını modellemeyi araştırmaktadır. 2016'dan 2018'e kadar uzanan Ulusal Güvenlik Açığı Veritabanı'ndan (NVD) alınan veriler ve Ortak Güvenlik Açığı Puanlama Sistemi (CVSS) puanları analiz edilmiştir. Çalışma, aylık güvenlik açığı sayıları ve ortalama aylık etki değerleri için Lojistik, Weibull, Nakagami, Gamma ve Log-lojistik dahil olmak üzere çeşitli dağıtım modellerini değerlendirir. Model sağlamlığı değerlendirmesi için uyum iyiliği testleri ve bilgi kriterleri uygulandı. Bulgular, araştırmacılar ve Android yazılım geliştiricileri için değerli içgörüler sunarak tahmin, risk değerlendirmesi, kaynak tahsisi ve araştırma yönüne yardımcı olur. Ortalama aylık etki değerleri ve aylık güvenlik açığı sayıları için sırasıyla lojistik ve Nakagami dağılımları en uygun modeller olarak ortaya çıkmıştır. Son olarak, istatistiksel yöntemler, anlaşılabilirlik, veri miktarı, hesaplama ihtiyacı ve veri bağımsızlığı gibi esnek özellikleri nedeniyle küçük veri kümeleri veya daha net tanımlanmış veriler için bilinen yapay zekâ yöntemlerine karşı daha iyi performans gösterir.

Anahtar Kelimeler: *İstatistiksel dağılımlar, Android güvenlik açıkları, Yazılım güvenliği, Güvenlik açığı keşif modeli.*

Article Info

Paper Type: Research Paper

Received: 31.07.2024

Accepted: 05.09.2024

Abstract

Android operating system is a mobile operating system that supports multimedia features. Android offers a wide range of applications and integrated features for playing, recording, editing and sharing audio, video, images and other multimedia content. Most Android devices include cameras, speakers, microphones, and other multimedia components. In software security, vulnerabilities are critical concerns that often emerge during software development. Predicting these vulnerabilities post-release is essential for risk assessment and mitigation. While various models have been explored, the Android operating system remains relatively uncharted. This study delves into modeling Android security vulnerabilities using different statistical distributions, comparing their suitability to the widely-used Alhazmi-Malaiya Logistic (AML) model. Data from the National Vulnerability Database (NVD) spanning 2016 to 2018, along with Common Vulnerability Scoring System (CVSS) scores, was analyzed. The study evaluates several distribution models, including Logistic, Weibull, Nakagami, Gamma, and Log-logistic, for monthly vulnerability counts and average monthly impact values. Goodness-of-fit tests and information criteria were applied for model robustness assessment. The findings offer valuable insights for researchers and Android software developers, aiding prediction, risk assessment, resource allocation, and research direction. Logistic and Nakagami distributions emerged as the best-fit models for average monthly impact values and monthly vulnerability counts, respectively. Finally, statistical methods perform better against known artificial intelligence methods for small data sets or more clearly defined data due to their flexible features such as comprehensibility, amount of data, need for calculation, and data independence.

Keywords: *Statistical distributions, Android vulnerabilities, Software security, Vulnerability discovery model*.

Atıf/ to Cite (APA): Gencer, K. & Basciftci, F. (2024). Modeling Android Security Vulnerabilities: Insights from Statistical Distributions. International Journal of Management Information Systems and Computer Science, 8(2), 110-126. DOI: 10.33461/uybisbbd.1524207

[1]"Ulusal yazılım açıklık veri tabanı oluşturulması kapsamında android açıklıklarının modellenmesi ve analiz edilmesi" isimli Selçuk Üniversitesinde Bilgisayar Mühendisliği alanında yapılan Doktora tezinden üretilmiştir.

[2] Dr. Öğretim Üyesi, Afyon Kocatepe Üniversitesi, Mühendislik Fakültesi, keremgencer09@hotmail.com, Afyonkarahisar, Türkiye.
[3] Prof. Dr., Selçuk Üniversitesi, Teknoloji Fakültesi, basciftci@selcuk.edu.tr, Konya, Türkiye.

1. INTRODUCTION

Knowing the security vulnerabilities inside the lifecycle of a software program might provide the means to evaluate, reduce, and even remove the risks these vulnerabilities produce. Though a software security growth model can be used during software creation, this may not stop security vulnerabilities from appearing in the software. Using a prediction model, vulnerability discovery prediction is as essential as vulnerability detection. Recently, many security vulnerability discovery models have been developed. In general, vulnerability discovery models are divided into time- and effort-based categories. In this study, the average monthly impact value and the monthly vulnerability count for Android are modeled for the first time. Until now, five probability distributions as flexible as the Weibull distribution: Normal, Logistic, Log-logistic, Nakagami, and Gamma distributions, have been tried alongside the Weibull distribution, which is often used in this kind of modeling, and their performances have been compared. These distributions are symmetrical and asymmetrical, i.e., skewed distributions. Probability density functions (pdf) of the distributions in question, cumulative distribution functions, goodness-of-fit tests, and measurement criteria were all compared to find the best model.

These results can guide researchers and software developers interested in Android vulnerabilities in several ways:

- Prediction and Risk Assessment: These results can be used to predict better the future impacts and probabilities of vulnerabilities on the Android operating system. This is important for developing strategies to combat vulnerabilities and better understanding potential risks.
- Software Development: Using these results, software developers can focus on safer coding practices for Android applications or operating systems. Applying a specific distribution or estimating the propagation rate of vulnerabilities can improve software security.
- Resource Allocation: These results can be used to allocate information security resources effectively. Understanding which vulnerabilities require more resources or further action can help use the budget more effectively.
- Research Direction: These results can help determine the direction of future security research. These findings about which statistical distributions better model a particular security environment can be a basis for future research. In this study, some advantages of statistical methods are as follows.
- Understandability: Statistical methods allow for a more straightforward interpretation of results. Therefore, it is more accessible to people who want to understand vulnerabilities and discover their causes. At the same time, these methods can determine more clearly which factors affect the likelihood of security vulnerabilities.
- Amount of Data: Statistical methods can work with more limited data. Complex AI methods such as deep learning often require large data sets, while statistical methods can deal with smaller data sets.
- Computational Need: Statistical methods may not extensively use computational resources such as deep learning. This can result in faster results at lower costs.
- Independence: Statistical methods are generally less data-dependent. They can be more flexible, especially when new data sets or updates arrive because retraining or adapting the model is less complex.

Some studies on vulnerability detection models are shown in Figure 1 (Movahedi, 2019). These studies are divided into two categories, time-based and effort-based, and the time-based models are further subdivided into three categories: Quasi-Linear, SGRM-based, and S-shape. It is understood that the focus of these studies was on time-based studies rather than effort-based studies.

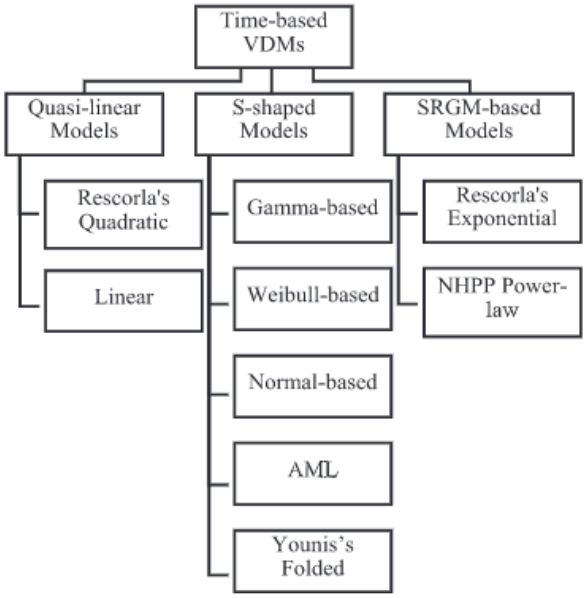

Figure 1. Taxonomy of Vulnerability Discovery Models

With this study – since the primary goal of modeling is to forecast – the most similar alternative distributions were compared. In Section 2, previous studies on the subject are investigated in detail. Section 3 introduces the distributions used in this study. Section 4 compares the goodness-of-fit tests often used to see how well the sample data matches the expected distribution values. In Section 5, information on the data set and the method of the study is given, and it presents the values obtained from the fitness measures on the proposed distributions. Sections 6 and 7 are the discussion and conclusion sections, respectively.

2. LITERATURE REVIEW

In general, discovery models are divided into time- and effort-based categories. While time-based models use time, effort-based ones use environmental factors such as CPU utilization and load count as the independent variables. Time-based models are more frequently studied. When these studies are investigated, it is seen that the first study proposed as a full-fledged vulnerability discovery model was Anderson's thermodynamics model (Anderson, 2002). However, this model was not sufficiently successful at detecting the weaknesses in various software. In later years, Alhazmi et al. conducted many studies on both time- and effort-based models (Alhazmi et al., 2005; Alhazmi and Malaiya, 2005a; 2005b; 2006a; 2006b; Alhazmi et al., 2007; Alhazmi and Malaiya, 2008). A statistical density-based model was developed by Rescorla (Rescorla, 2005). Woo et al. attempted to create a vulnerability discovery model on three popular web browsers (Woo et al., 2006a). They concluded that the model will be fixed when categorized according to the severity and the type of vulnerabilities. Also, Woo et al. conducted a study investigating Apache and IIS web server vulnerabilities (Woo et al., 2006b). Kim et al. proposed a model that searches for vulnerabilities in different software versions (Kim et al., 2007). Joh et al. proposed a Weibull distribution-based model that can be used when asymmetrical data sets (Joh et al., 2008). Chen et al. proposed a vulnerability discovery model that used a multi-loop method (Chen et al., 2010). Woo et al. observed that models cannot make good predictions if the obtained data does not feature trend changes (Woo et al., 2011). Ozment conducted a study on the limitations of vulnerability discovery models (Ozment, 2007), while Massacci and Nguyen investigated the available vulnerability discovery models in terms of quality and predictability (Massacci and Nguyen, 2014). Anand and Bhatt studied convex-shaped discovery models using five parameters and the weighted criterion method (Anand and Bhatt, 2016). Anand et

al. also developed a model for multi-version software (Anand et al., 2017). Bhatt et al. conducted a study on the relationship between vulnerabilities discovered recently and vulnerabilities found in the past (Bhatt et al., 2017). Kansal et al. developed a model that links the number of commercial software users (Kansal et al., 2018). In another study, Kansal et al. investigated the relationship between the operational coverage function and the expected vulnerability count with a generalized statistical model (Kansal et al., 2017). Johnston used the Bayesian method in their Ph.D. thesis on vulnerability discovery modeling (Johnston, 2018). Again, Johnston et al. conducted a study that connected the software release date and the security evaluation profile (Johnston et al., 2018). Rahimi and Zargham developed a model on code complexity and quality that does not require past vulnerability data (Rahimi and Zargham, 2013); however, since it was not possible to get the complete source code – as in other studies – the model could not be put to general use. Scandariato and Walden studied Android application vulnerabilities using support vector machines (Scandariato and Walden, 2012).

This study utilized source code and could only be used in open-sourced applications and was therefore limited because it couldn't be used on closed-source applications. Scandariato et al. used text mining in their studies on open-sourced Android applications (Scandariato et al., 2014). In their study, Gencer and Başçiftçi (2021) propose a model called F-CVSS (Fuzzy Common Vulnerability Scoring System) by combining fuzzy logic and logistic regression as an alternative to the traditional CVSS (Common Vulnerability Scoring System) system. They attempted to determine the relevant components with their investigations, and this method was successful in determining the appropriate features and finding vulnerabilities in them. Younis et al. investigated and modeled cases where the vulnerabilities occur asymmetrically (Younis et al., 2011). Wang et al. proposed an effort-based model, and they claimed that this model achieved better results than AML (Wang et al., 2019). Finally, Pokhrel et al. used time series, Artificial Neural Networks (ANNs), and Support Vector Machines (SVMs) to investigate desktop operating systems (Pokhrel et al., 2017). In their article, Gencer and Başçiftçi (2021) use ARIMA and deep learning methods to perform a time series analysis of vulnerabilities in the Android operating system. The study compares various time series modeling techniques to predict future trends of these vulnerabilities and to identify possible risks in advance. Movahedi et al. introduced an approach for predicting the cumulative number of software vulnerabilities with a neural network model. (Movahedi et al., 2019).

3. LIFECYCLE DISTRIBUTIONS USED IN MODELING ANDROID SOFTWARE VULNERABILITIES

Life analysis is the collection of all the statistical techniques used to analyze the data gathered while the model above was being created. Life analysis data sets are usually represented by classical statistical distributions such as Exponential, Gamma, Weibull, Log-normal, and Logistic (Nelson, 1982; Lawless, 2003; Lee and Wenyu, 2003; Kleinbaum and Klein, 2005; Machin et al., 2006). This section introduces lifecycle distributions, such as the Weibull, Gamma, Logistic, Log-logistic, Normal, and Nakagami, used to model the monthly counts and average monthly impact scores of Android vulnerabilities between 2016 and 2018. These flexible distributions are popularly used in reliability theory and adapt to many data sets.

3.1. Weibull Distribution

The Weibull distribution was proposed in 1939 by the physicist Waloddi Weibull, who gave his name to the distribution. As a flexible distribution, it is often used in engineering applications and modeling compounds, i.e., random variables. The Weibull distribution is also used in electronic circuits and to observe some biological organisms' decay rates. At the beginning of the 1970s, it began to be used in seismic risk analysis. The Weibull distribution became famous thanks to its usability in cases where the variable has a positive value, such as applications in the financial sector. Its probability density function $f(x)$ and distribution function $F(x)$ are given in Equations (1) and (2), respectively.

$$f(x)=\frac{\gamma}{\lambda}\left(\frac{x}{\lambda}\right)^{\gamma-1} e^{-\left(\frac{x}{\lambda}\right)^{\gamma}}, \quad \lambda,\gamma,x>0 \tag{1}$$

$$F(x)=\left(1-e^{-\lambda x}\right)^{\gamma} \tag{2}$$

Here, *x*, *γ,* and *λ* are the random variables representing the monthly average score (or the monthly vulnerability count), shape, and scale parameters.

3.2. Log-logistic Distribution

Log-logistic distribution is one of the alternatives to Weibull, a distribution with two parameters. If *Log (T)* has a logistic distribution, the lifecycle *T* has a log-logistic distribution. This distribution successfully models data with tremendous and small values in some example series (Ahmad et al., 1988; Kantam et al., 2001). It is more successful than the Log-normal distribution in time series data with sudden changes (Shoukri et al., 1988). Its probability density function *f(x)* and distribution function *F(x)* are given in Equations (3) and (4), respectively.

$$f(x)=\frac{\gamma}{\lambda}\frac{\left(\frac{x}{\lambda}\right)^{\gamma-1}}{\left(1+\left(\frac{x}{\lambda}\right)^{\gamma}\right)^{2}} \tag{3}$$

$$F(x)=\frac{1}{\left(1+\left(\frac{x}{\lambda}\right)^{\gamma-1}\right)}, \quad \mathrm{x},\gamma,\lambda>0 \tag{4}$$

Here, *x* and *λ* are the random variables representing the monthly average score (or the monthly vulnerability count), shape, and scale parameters, respectively.

3.3. Normal Distribution

Also known as the Gaussian distribution, the Normal distribution has practical applications in many areas. It is an essential continuous probability distribution family (Hogg and Craig, 1978). The Normal distribution has two parameters: the arithmetic mean, μ, and the variance, σ^2. The probability density function *f(x)* is shown in Equation (5) (Casella and Berger, 2001):

$$f(x)=\frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{(x-\mu)^2}{2\sigma^2}}, \quad \mathrm{x}\in\mathbb{R} \tag{5}$$

Here, *x* is the random variable representing the monthly average score or vulnerability count.

3.4. Gamma Distribution

This continuous probability distribution is used in probability theory and statistics using two parameters. The Gamma distribution is used to model the size of insurance demand and rainfall (Anderson and Darling, 1954; Boland, 2007). Its probability density function *f(x)* is given in Equation (6):

$$f(x) = x^{k-1}\frac{e^{-\frac{x}{\theta}}}{\theta^{k}\Gamma(k)}, \quad x>0, \quad k,\theta>0 \tag{6}$$

Here, *x*, *k*, and λ are the random variables representing the monthly average score (or the monthly vulnerability count), shape, and scale parameters, respectively.

3.5. Logistic Distribution

The Logistic distribution is a continuous probability distribution. It is similar to the bell-curved Normal distribution in terms of shape, but it is flatter due to the larger weights at the tails. Its probability density function is known to be the square of a hyperbolic secant function (Decani and Stine, 1986). Its probability density function *f(x)* is given in Equation (7):

$$f(x) = \frac{e^{-\frac{(x-\mu)}{s}}}{s\left(1+e^{-\frac{(x-\mu)}{s}}\right)^{2}}, \quad x \in \mathbb{R} \tag{7}$$

Here, *x*, μ, and *s* are the random variables representing the monthly average score (or the monthly vulnerability count), location, and scale parameters, respectively.

3.6. Nakagami Distribution

The Nakagami distribution is commonly used to model right-skewed data sets with positive values. Though there have been many distributions that model radio signal weaknesses, such as Weibull and Log-normal, in 1960, Nakagami proposed this distribution instead (Nakagami, 1960). The Nakagami distribution has been the main focus of some studies thanks to its wide applicability compared to other popular distribution models (Türksen et al., 2015), and it is used in various areas. It has been observed to exhibit good performance in generating unit hydrographs used to predict flow rates in hydrology by Sarkar, Goel, and Mathur (Sarkar et al., 2009; 2010). Shankar et al. and Tsui et al. used it in medical imaging and for modeling ultrasound data, respectively (Shankar et al., 2005; Tsui et al., 2006). Kim and Latchman analyzed motion picture data using the Nakagami distribution (Kim and Latchman, 2009). Furthermore, Nakahara and Carcole showed the usability of the Nakagami distribution in seismic study modeling (Nakahara and Carcolé, 2010). The probability density function *f(x)* of the Nakagami distribution is given in Equation (8):

$$f(x) = \frac{2m^{m}}{\Gamma(m)\Theta^{m}}x^{2m-1}\exp\left(mlogx^{2} - \frac{mx^{2}}{\Theta}\right)x^{-1} \tag{8}$$

Here, $x>0$, *m*, and Θ are the random variables representing the monthly average score (or the monthly vulnerability count), location, and scale parameters, respectively. In mathematics, the gamma function (Γ) is the generalization of the factorial function for complex and non-integer real numbers.

4. MODEL FITTING AND GOODNESS-OF-FIT ANALYSES

This section introduces three goodness-of-fit tests, which will be used to measure distribution fitness: Kolmogorov-Smirnov, Anderson-Darling, and Cramer-von Mises.

4.1. Kolmogorov-Smirnov GIoodness-of-Fit Test

One of the goodness-of-fit tests used in this study is the Kolmogorov-Smirnov test (Kolmogorov, 1933). This tests the fitness of a data set on a statistical model. It is a method used

successfully among goodness-of-fit tests based on an experimental distribution function. It is known as the Kolmogorov-Smirnov (*K-S*) goodness-of-fit test in literature because Kolmogorov developed it but was first used in goodness-of-fit tests by Smirnov. In the K-S test, with *x* number of samples, the cumulative distribution function $F_O(x)$ is determined, which is assumed to be a fixed distribution. $S_n(x)$ is the experimental cumulative distribution function that gives the ratio of the values that are smaller than, or equal to, a value *x* across *n* observed samples. According to the main idea of the K-S test, if the experimental distribution function results are not close enough to the hypothetical $F_O(x)$ value, it is deduced that the observed data does not follow the theoretical distribution. In other words, the observed data do not fit the claimed distribution. The statistic to test this condition is shown below:

$$D = \max_x \left| F_0^*(X) - S_n(x) \right| \tag{9}$$

Where *x* is the sample count. The *D* statistic of Kolmogorov and Smirnov is entirely independent of the hypothetical distribution under test when $F_O(x)$ is continuous and fully known (Kolmogorov, 1933; Smirnov, 1939). The distribution of this statistic can be obtained when all the parameters are known. Otherwise, there is no distribution of the *D* statistic.

4.2. Anderson-Darling Goodness-of-Fit Test

Anderson and Darling proposed another test statistic by adapting the K-S test (Anderson and Darling, 1954). To determine this statistic, *n* unit samples $\{X_1, X_2, .. X_n\}$ are drawn from a batch whose probability function and probability function parameters are known. The null hypothesis for the Anderson-Darling test is built on the assumption that the samples come from a distribution determined entirely by the parameters. If the null hypothesis is rejected due to the test, it is deduced that the data do not fit the distribution determined by the parameters. This test was not created for specific distributions but all distributions whose parameters are known. Later, it was improved for cases with unknown parameters.

The Anderson-Darling test statistic is shown in Equation (10) where *x*, $F_O(x)$, and *i* are the sample count, the cumulative distribution function, which is assumed to be fixed, and the rank value.

$$A^2 = -\frac{2}{n}\sum_{i=1}^{n}\left[\begin{array}{l}\left(i-\frac{1}{2}\right)\log\left\{F_0\left(x_{(i)}\right)\right\} \\ +\left(n-i+\frac{1}{2}\right)\log\left\{1-F_0\left(x_{(i)}\right)\right\}\end{array}\right] - n \tag{10}$$

4.3. Cramer von Mises Goodness of Fit Test

The Cramer-von Mises goodness-of-fit test was proposed by Harald Cramer and Richard Edler Mises (Cramér, 1928). The Cramer-von Mises (W_n) test statistic is defined as follows:

$$W_n = \sum_{i=1}^{n}\left\{F_0\left(x_{(i)} - \frac{2i-1}{2n}\right)\right\}^2 + \frac{1}{12n} \tag{11}$$

Where *x*, *n*, $F_O(x)$ and *I* are the sample count, the random sample $\{X_1, X_2, .. X_n\}$, the cumulative distribution function, which is assumed to be fixed, and the

If the test statistic obtained for the observed value is larger than the table value, it shows that the data do not follow the distribution proposed.

5. DATA AND METHODOLOGY

The vulnerability data used in this study is taken from the National Vulnerability Database (NVD), the largest source in this area (NVD, 2019). NVD is a large-scope database formed by data gathered from companies located inside and outside America, with contributions from the United States government. It is the most preferred database in terms of its policies on widespread use and

public availability. The American National Security Agency supports the NVD project. Vulnerabilities announced by NVD receive a Common Vulnerability and Exposures (CVE) number. Hence, different numbers and re-announcements for the same exposure are prevented. The Android vulnerabilities were filtered out while the database was being formed (Cvedetails, 2019). Furthermore, Common Vulnerability System Scores (CVSSs) for Android vulnerabilities between the specified dates are grouped every month. The study goal was to model the monthly impact scores and the monthly vulnerability counts. After the data was gathered, Weibull, Logistic, Normal, Log-logistic, and Nakagami distributions were applied to obtain the monthly vulnerability impact scores. The goodness-of-fitness test results of these distributions are given in Table 1.

Table 1. National Vulnerability Database Average Score Goodness of Fits

Goodness of Fits	Distributions				
	Weibull	**Logistic**	**Normal**	**Log-logistic**	**Nakagami**
K-S Statistics	0.1461	**0.0985**	0.1427	0.1078	0.1502
A-D Statistics	1.0423	**0.5936**	0.9677	0.7442	1.0603
CVM Statistics	0.1851	**0.0821**	0.1604	0.0944	0.1763
K-S (*p*-value)	0.4263	**0.8757**	0.4561	0.7972	0.3911
A-D (*p*-value)	0.3353	**0.6529**	0.3740	0.5220	0.3266
CVM (*p*-value)	0.2992	**0.6828**	0.3607	0.6158	0.3195

It was observed that the *p*-values of all the distributions under investigation were larger than 0.05. However, the purpose of this study was not just to find the distributions that model the monthly average scores but to find the distribution that models it best (*p*-value>0.05). Nevertheless, according to the Kolmogorov-Smirnov, Anderson-Darling and Cramer-von Mises test statistics, the best distribution is observed to be the Logistic distribution (*p*-value>0.05). Furthermore, Weibull, Logistic, Log-logistic, Gamma and Nakagami distributions were applied to model the monthly vulnerability counts. The data on these distributions are given in Table 2.

Table 2. National Vulnerability Database Monthly Count Goodness of Fit

Goodness of Fit	Distributions				
	Weibull	**Logistic**	**Log-logistic**	**Gamma**	**Nakagami**
K-S Statistics	0.1085	0.1412	0.1058	0.1302	**0.1137**
A-D Statistics	0.2817	0.7853	0.6940	0.4537	**0.2782**
CVM Statistics	0.0457	0.1316	0.0753	0.0775	**0.0475**
K-S (*p*-value)	0.7908	0.4697	0.8149	0.5748	**0.7409**
A-D (*p*-value)	0.9509	0.4908	0.5628	0.7932	**0.9533**
CVM (*p*-value)	0.9044	0.4525	0.7226	0.7097	**0.8942**

The most successful distribution was identified by looking at the Kolmogorov-Smirnov, Anderson-Darling and Cramer-von Mises test statistics and the *p*-values. It was observed that the *p*-values of all the distributions under investigation were larger than 0.05. However, the purpose of this study was not just to find the distributions that model the monthly average scores but to find the distribution that models it best (*p*-value>0.05). Accordingly, the Kolmogorov-Smirnov, Anderson-Darling and Cramer-von Mises test statistics indicated that the best distribution was seen to be the Nakagami distribution (*p*-value>0.05).

5.1. Comparison of Vulnerability Discovery Models

The −2Log L statistic is one of the metrics used to decide a suitable lifecycle model. The most suitable model is the model with the lowest value (Klein and Moeschberger, 1997). Akaike proposed the Akaike Information Criterion (AIC) to compare different models, and this is defined as follows:

$$AIC = -2lnL+2k \tag{12}$$

where k is the number of model parameters (Akaike, 1974).

In this equation, *lnL* and *k* are the log-likelihood and the parameter count, respectively. The smallest AIC value is used to decide the best model (Cavanaugh, 1997). Another information criterion that is widely used in literature is the Bayesian or Schwarz Information Criterion (BIC). This is defined as follows (McLachlan and Peel, 2001):

$$\mathrm{BIC} = -2lnL + k\,(\log(n)) \tag{13}$$

In this equation, *ln(n)* is the natural logarithm of the sample volume *n*. The *k* and *n* symbols represent the number of parameters and the sample size. Again, the smallest BIC value is used to decide the best model (Hurvich and Tsai, 1989; Cavanaugh, 1997; Ucal, 2006). In Figure 2, the pdf of the best distributions that model the monthly average scores of NVD are given. It is seen that the best distribution is the Logistic distribution. After that, the sequence of the most suitable distributions in terms of fitness was Log-logistic, Normal, Nakagami and Weibull.

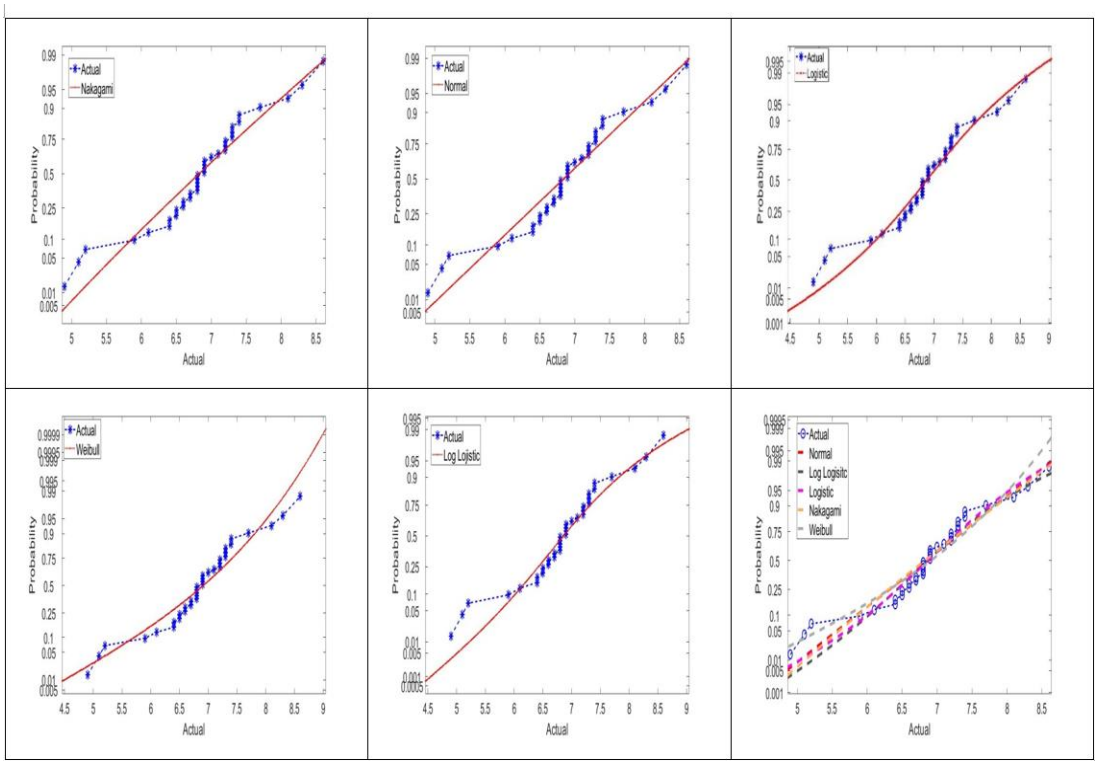

Figure 2. National Vulnerability Database Average Score Probability Density Functions

Furthermore, the cumulative distribution functions are given in Figure 3 below.

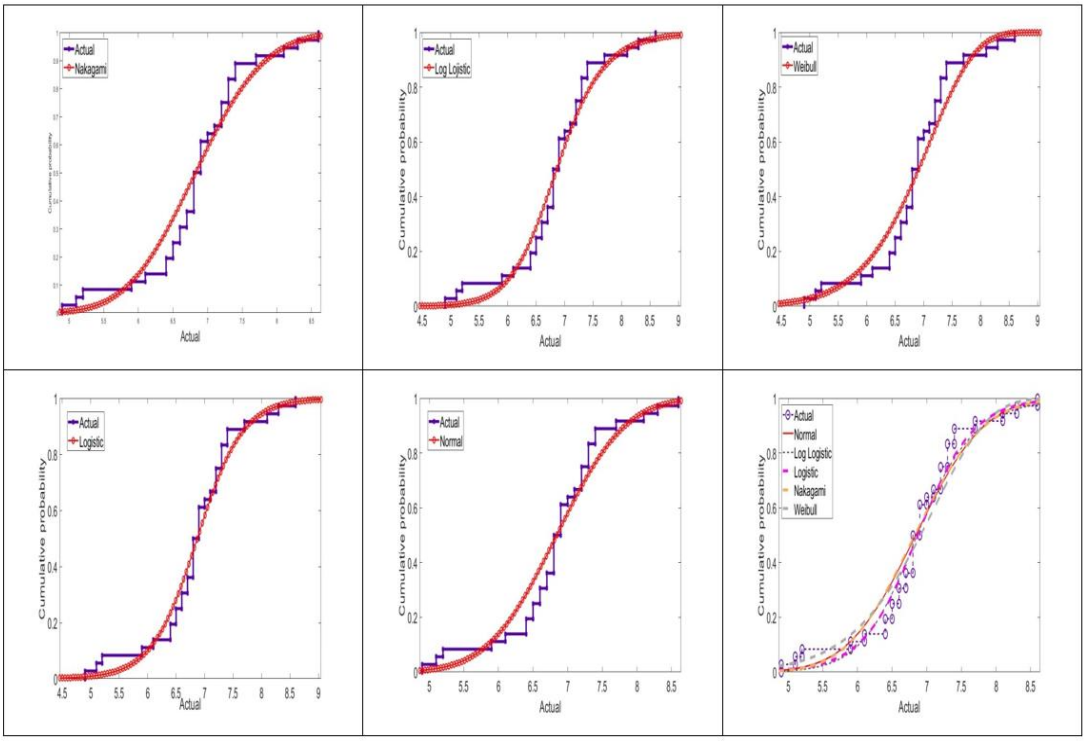

Figure 3. National Vulnerability Database Monthly Count Cumulative Distribution Functions

In Table 3, known model fitting metrics, AIC, BIC and −2Log*L* have been used for model prediction. According to these metrics, the Logistic distribution had the smallest AIC, BIC and −2Log*L* values. After that, the sequence of the remaining distributions was Log-logistic, Normal, Nakagami and finally Weibull in terms of their values.

Table 3. National Vulnerability Database Average Score Model Fitting

Model Fitting	**Distributions**				
	Weibull	**Logistic**	**Normal**	**Log-logistic**	**Nakagami**
LogL	−41.7878	**−39.9638**	−41.3863	−40.8412	−41.7680
−2LogL	83.5755	**79.9275**	82.7726	81.6823	83.5360
AIC	87.5755	**83.9275**	86.7726	85.6823	87.5360
BIC	90.7426	**87.0945**	89.9396	88.8494	90.7030

In Table 4, known model fitting metrics, AIC, BIC and −2Log*L* have been used to find the model that best predicts the monthly vulnerability counts. According to these metrics, the Nakagami and Logistic distributions have the smallest and the largest AIC, BIC and −2Log*L* values, respectively.

Table 4. National Vulnerability Database Monthly Count Model Fitting

Model Fitting	**Distributions**				
	Weibull	**Logistic**	**Log-logistic**	**Gamma**	**Nakagami**
Log*L*	−173.6286	−175.9572	−176.3994	−174.2972	**−173.6244**
−2Log*L*	347.2572	351.9144	352.7989	348.5945	**347.2487**
AIC	351.2572	355.9144	356.7989	352.5945	**351.2487**
BIC	354.4242	359.0814	359.9659	355.7615	**354.4157**

In Table 5, the parameters, the lower and upper bounds of these parameters and the standard errors of these parameters are given for the Weibull, Logistic, Normal, Log-logistic and Nakagami distributions that model the monthly average impact values. With the parameter values obtained, the monthly average impact value – determined as a random variable – can be predicted.

Table 5. National Vulnerability Database Average Score Parameters

Parameters	**Distributions**				
	Weibull	**Logistic**	**Normal**	**Log-logistic**	**Nakagami**
α	7.1734	**6.8721**	6.8417	6.8607	19.7059
β	9.8417	**0.3978**	0.7639	16.8118	47.3919
Lower Bound (α)	6.9218	**6.6529**	6.5921	6.6368	10.6784
Lower Bound (β)	7.5001	**0.2850**	0.5875	12.0231	43.9045
Upper Bound (α)	7.4250	**7.0913**	7.0912	7.0845	28.7333
Upper Bound (β)	12.1833	**0.5107**	0.9403	21.6004	50.8794
Standard Error (α)	0.1284	**0.1118**	0.1273	0.1142	4.6059
Standard Error (β)	1.1947	**0.0576**	0.0900	2.4432	1.7793

α: The shape parameter

β: The scale parameter

In Table 6, the parameters, the lower and upper bounds for these parameters, and the standard errors for these parameters are given for the Weibull, Logistic, Log-logistic, Gamma and Nakagami distributions that model the monthly vulnerability counts. With the parameter values obtained, the monthly vulnerability count – determined as a random variable – can be predicted.

Table 6. National Vulnerability Database Monthly Count Parameters

Parameters	**Distributions**				
	Weibull	**Logistic**	**Log-logistic**	**Gamma**	**Nakagami**
α	61.5340	52.4923	47.9460	2.3463	**0.7845**
β	1.7183	17.9225	2.4614	23.4290	**3948.1608**
Lower Bound (α)	49.2384	42.3484	37.0752	1.3300	**0.4722**
Lower Bound (β)	1.2800	13.0212	1.7724	12.1173	**2537.3748**
Upper Bound (α)	73.8297	62.6361	58.8169	3.3627	**1.0969**
Upper Bound (β)	2.1565	22.8237	3.1504	34.7408	**5358.9468**
Standard Error (α)	6.2734	5.1755	5.5465	0.5185	**0.1594**
Standard Error (β)	0.2236	2.5007	0.3516	5.7714	**719.8020**

α: The shape parameter

β: The scale parameter

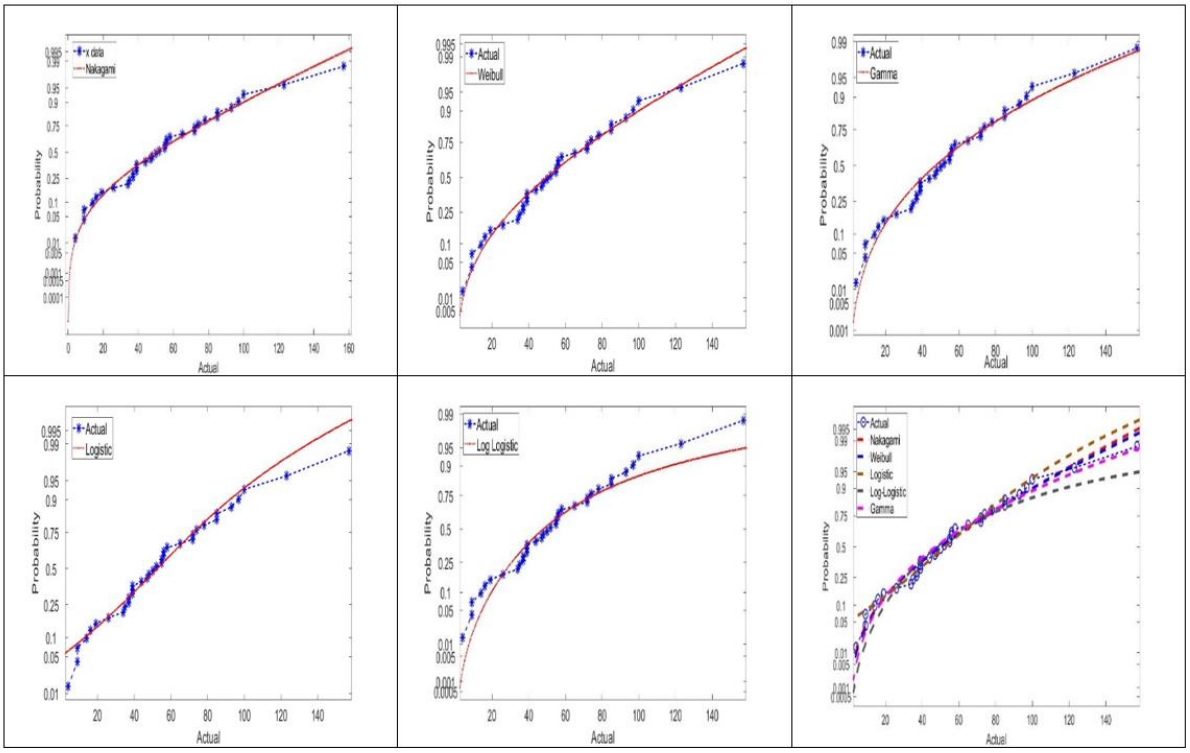

Figure 4. National Vulnerability Database Monthly Count Probability Density Functions

Figure 4 shows the pdf of the distributions that best model the monthly counts of NVD. It is seen that the best distribution is the Nakagami, followed by the Weibull distribution.

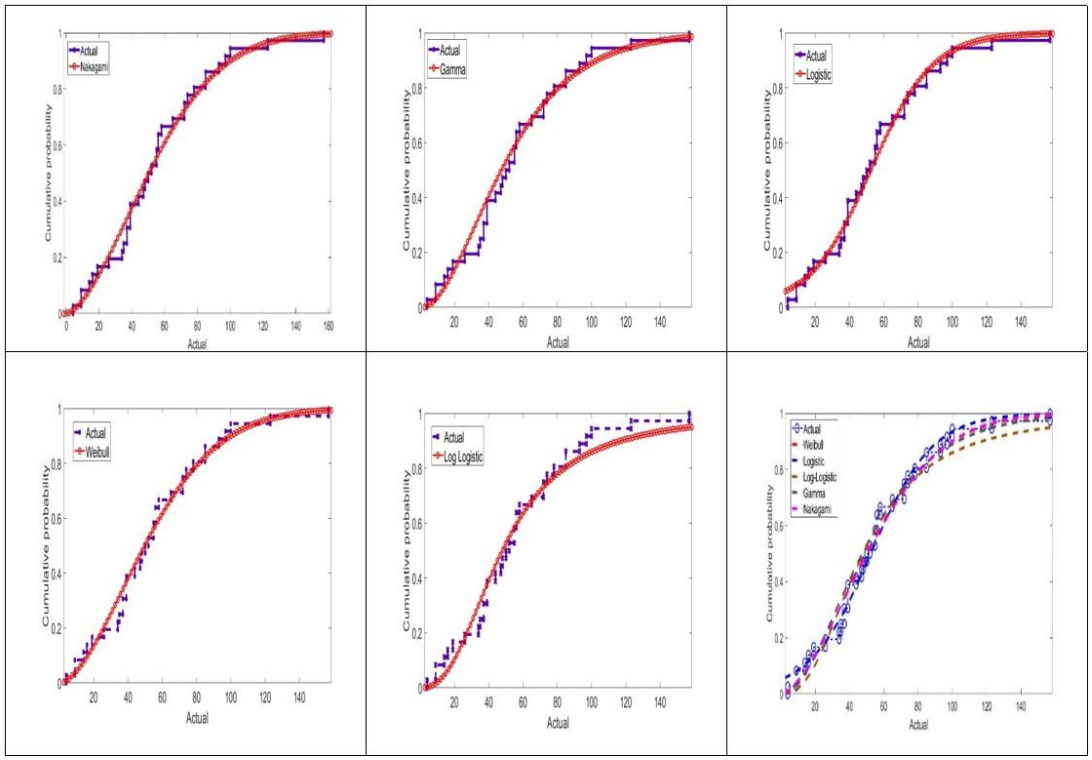

Figure 5. National Vulnerability Database Monthly Average Score Cumulative Distribution Functions

After those, the remaining distribution sequence is Gamma, Logistic and Log-logistic in terms of fitness. Furthermore, the cumulative distribution functions are given in Figure 5.

6. DISCUSSION

This study investigated the symmetrical and asymmetrical Weibull, Logistic, Log-logistic, Normal, Gamma and Nakagami distributions that best model the monthly Android vulnerability scores and monthly vulnerability counts. In Table 1, showing the fitness comparisons of monthly average vulnerability score models, the largest (K-S) p values were 0.8757 and 0.3911 for the Logistic and Nakagami distributions, which are the best and the worst models, respectively. In Table 3, the $-2\text{Log}L$, AIC and BIC values for the Logistic distribution were observed to be 79.9275, 83.9275 and 87.0945, respectively. In Table 2, showing the fitness comparisons of monthly vulnerability count models, the largest (K-S) p values were 0.7409 and 0.4697 for the Nakagami and Logistic distributions, which are the best and the worst models, respectively. In Table 3, −2LogL, AIC and BIC values for the Nakagami distribution were observed to be 347.2487, 351.2487 and 354.4157, respectively. While the monthly average vulnerability scores were observed to be symmetrical, monthly vulnerability counts were observed to be more skewed. With the help of certain criteria, the predictive ability and fitness of these six distributions were compared. The predictive ability was measured by calculating the smallest AIC, BIC and 2LogL values. For the fitness criteria, Kolmogorov-Smirnov, Anderson-Darling and Cramer-von Mises statistical tests were applied. All the distributions were observed to represent the data sets well. However, the Logistic and Nakagami distributions were observed to best model the monthly average vulnerability scores and monthly vulnerability counts.

7. CONCLUSION

Vulnerability detection models help us forecast software vulnerabilities and enable the necessary precautions to be taken, such as planning the generation of a patch. In this study, the best distribution was determined by modeling the continuous vulnerabilities of the Android operating system from 2016 to 2018 with different statistical distributions. As a result of this study, it was seen that data sets usually modeled with Weibull distribution in published literature can also be modeled with different distributions. Android vulnerabilities have been best modeled by Logistic and Nakagami distributions for average monthly scores and monthly Android vulnerability counts, respectively. Goodness-of-fit tests have shown the fitness of these distributions. This study has proposed a new aspect of time-based vulnerability discovery models regarding statistical distributions. With suitable distributions, it has been shown that Android vulnerabilities can be modeled, and forecasts can be made. This study shows that these distributions are promising for accurate predictions of software vulnerability disclosures and results are helpful in academia and industry. As a result, it is aimed that analysts prioritize their work by taking into account the severity of the potential risks arising from Android vulnerabilities.

REFERENCES

Ahmad, M. I., Sinclair, C. D. and Werritty, A., 1988, Log-Logistic Flood Frequency Analysis, *Journal Of Hydrology*, 98 (3), 205-224.

Akaike, H., 1974, A New Look At The Statistical Model Identification, *Ieee Transactions On Automatic Control*, 19 (6), 716-723.

Alhazmi, O., Malaiya, Y. Ve Ray, I., 2005, Security Vulnerabilities In Software Systems: A Quantitative Perspective, *Data And Applications Security Xix*, Berlin, Heidelberg, 281-294.

Alhazmi, O. H. and Malaiya, Y. K., 2005a, Modeling The Vulnerability Discovery Process, *16th Ieee International Symposium On Software Reliability Engineering (Issre'05)*, Ten Pp.-138.

Alhazmi, O. H. and Malaiya, Y. K., 2005b, Quantitative Vulnerability Assessment Of Systems Software, *Annual Reliability And Maintainability Symposium, 2005. Proceedings*, 615-620.

Alhazmi, O. H. and Malaiya, Y. K., 2006a, Measuring And Enhancing Prediction Capabilities Of Vulnerability Discovery Models For Apache And Iis Http Servers, *17th International Symposium On Software Reliability Engineering*, 343-352.

Alhazmi, O. H. and Malaiya, Y. K., 2006b, Prediction Capabilities Of Vulnerability Discovery Models, *Rams '06. Annual Reliability And Maintainability Symposium, 2006.*, 86-91.

Alhazmi, O. H., Malaiya, Y. K. and Ray, I., 2007, Measuring, Analyzing And Predicting Security Vulnerabilities In Software Systems, *Computers & Security*, 26 (3), 219-228.

Alhazmi, O. H. and Malaiya, Y. K., 2008, Application Of Vulnerability Discovery Models To Major Operating Systems, *Ieee Transactions On Reliability*, 57 (1), 14-22.

Anand, A. and Bhatt, N., 2016, Vulnerability Discovery Modeling And Weighted Criteria Based Ranking, *Journal Of The Indian Society For Probability And Statistics*, 17 (1), 1-10.

Anand, A., Das, S., Agrawal, D. Ve Klochkov, Y., 2017, Vulnerability Discovery Modelling For Software With Multi-Versions, In: Advances In Reliability And System Engineering, Eds: Ram, M. Ve Davim, J. P., *Cham*: Springer International Publishing, P. 255-265.

Anderson, R., 2002, Security In Open Versus Closed Systems -The Dance Of Boltzmann, Coase And Moore, *Open Source Software Economics*, 127-142.

Anderson, T. W. and Darling, D. A., 1954, A Test Of Goodness Of Fit, *Journal Of The American Statistical Association*, 49 (268), 765-769.

Bhatt, N., Anand, A., Yadavalli, V. S. S. and Kumar, V., 2017, Modeling And Characterizing Software Vulnerabilities, *International Journal Of Mathematical, Engineering And Management Sciences*, 2 (4), 288-299.

Boland, P. J., 2007, Statistical And Probabilistic Methods In Actuarial Science, *Usa*, Taylor & Francis Inc, P. 43.

Casella, G. and Berger, R. L., 2001, Statistical Inference *Usa*, Duxbury, P. 102.

Cavanaugh, J. E., 1997, Unifying The Derivations For The Akaike And Corrected Akaike Information Criteria, *Statistics & Probability Letters*, 33 (2), 201-208.

Chen, K., Feng, D.-G., Su, P.-R., Nie, C.-J. and Zhang, X.-F., 2010, Multi-Cycle Vulnerability Discovery Model For Prediction, *Journal Of Software*, 21 (9), 2367-2375.

Cramér, H., 1928, On The Composition Of Elementary Errors, *Scandinavian Actuarial Journal*, 1928 (1), 141-180.

Cvedetails, 2019, https://www.cvedetails.com/browse-by-date.php, [Accessed Date: 10 June 2024].

Decani, J. S. and Stine, R. A., 1986, A Note On Deriving The Information Matrix For A Logistic Distribution, *The American Statistician*, 40 (3), 220-222.

Gencer, K. and Başçiftçi, F. 2021, Time series forecast modeling of vulnerabilities in the android operating system using ARIMA and deep learning methods. *Sustainable Computing: Informatics and Systems*, *30*, 100515.

Gencer, K. and Başçiftçi, F. 2021, The fuzzy common vulnerability scoring system (F-CVSS) based on a least squares approach with fuzzy logistic regression. *Egyptian Informatics Journal*, *22*(2), 145-153.

Hogg, R. V. and Craig, A. T., 1978, Introduction To Mathematical Statistics *Newyork*, Macmillan, P. 109.

Hurvich, C. M. and Tsai, C.-L., 1989, Regression And Time Series Model Selection In Small Samples, *Biometrika*, 76 (2), 297-307.

Joh, H., Kim, J. and Malaiya, Y. K., 2008, Vulnerability Discovery Modeling Using Weibull Distribution, *2008 19th International Symposium On Software Reliability Engineering (Issre)*, 299-300.

Johnston, R., 2018, A Multivariate Bayesian Approach To Modeling Vulnerability Discovery In The Software Security Lifecycle, Ph.D, *George Washington University*, Washington, Dc, Usa, 55-65.

Johnston, R., Sarkani, S., Mazzuchi, T., Holzer, T. and Eveleigh, T., 2018, Multivariate Models Using Mcmcbayes For Web-Browser Vulnerability Discovery, *Reliability Engineering & System Safety*, 176, 52-61.

Kansal, Y., Kapur, P. K., Kumar, U. and Kumar, D., 2017, User-Dependent Vulnerability Discovery Model And Its Interdisciplinary Nature, *Life Cycle Reliability And Safety Engineering*, 6 (1), 23-29.

Kansal, Y., Kapur, P. K. and Kumar, U., 2018, Coverage-Based Vulnerability Discovery Modeling To Optimize Disclosure Time Using Multiattribute Approach, *Quality And Reliability Enginering International*, 35 (1), 62-73.

Kantam, R. R. L., Rosaiah, K. and Rao, G. S., 2001, Acceptance Sampling Based On Life Tests: Log-Logistic Model, *Journal Of Applied Statistics*, 28 (1), 121-128.

Kim, J., Malaiya, Y. K. and Ray, I., 2007, Vulnerability Discovery In Multi-Version Software Systems, *10th Ieee High Assurance Systems Engineering Symposium (Hase'07)*, 141-148.

Kim, K. and Latchman, H. A., 2009, Statistical Traffic Modeling Of Mpeg Frame Size: Experiments And Analysis. Journal Of Systemics, *Cybernetics And Informatics*, 7 (6), 54-59.

Klein, J. P. and Moeschberger, M. L., 1997, Survival Analysis Techniques For Censored And Truncated Data, *Newyork*, Springer, P. 277.

Kleinbaum, D. G. and Klein, M., 2005, Survival Analysis: A Self-Learning Text, *Usa*, Springer, P. 590.

Kolmogorov, A. N., 1933, Sulla Determinazone Empirica Di Une Legge Di Distribuzione, *G. İst. Attuari*, 83-91.

Lawless, J. F., 2003, Statistics Models And Methods For Lifetime Data, *New Jersey*, John Wiley & Sons, P. 630.

Lee, E. T. and Wenyu, J. W., 2003, Statistical Methods For Survival Data Analysis, *Newyork*, John Wiley &Sons, P. 513.

Machin, D., Cheung, Y. B. and Parmar, M., 2006, Survival Analysis: A Practical Approach, *England*, John Wiley & Sons, P. 266.

Massacci, F. and Nguyen, V. H., 2014, An Empirical Methodology To Evaluate Vulnerability Discovery Models, *Ieee Transactions On Software Engineering*, 40 (12), 1147-1162.

Mclachlan, G. and Peel, D., 2001, Finite Mixture Model, *Newyork*, Wiley, P. 419.

Movahedi, Y., Cukier, M. and Gashi, I., 2019, Vulnerability Prediction Capability: A Comparison Between Vulnerability Discovery Models And Neural Network Models, *Computers & Security*, 87, 1-10.

Nakagami, M., 1960, The M-Distribution—A General Formula Of Intensity Distribution Of Rapid Fading, In: Statistical Methods In Radio Wave Propagation, Eds: Hoffman, W. C.: Pergamon, P. 3-36.

Nakahara , H. and Carcolé, E., 2010, Maximum-Likelihood Method For Estimating Coda Q And The Nakagami-M Parameter, *Bulletin Of The Seismological Society Of America*, 100 (6), 3174-3182.

Nelson, W. B., 1982, Applied Life Data Analysis, *Canada*, John Wiley & Sons, P. 634.

Nvd, 2019,https://nvd.nist.gov/ [Accessed Date: 10 June 2024].

Ozment, A., 2007, Improving Vulnerability Discovery Models. Proceedings Of The 2007 Acm Workshop On Quality Of Protection. Alexandria, Virginia, Usa, Acm**:** 6-11.

Pokhrel, N. R., Rodrigo, H. and Tsokos, C. P., 2017, Cybersecurity: Time Series Predictive Modeling Of Vulnerabilities Of Desktop Operating System Using Linear And Non-Linear Approach, 8 (4), 362-382.

Rahimi, S. and Zargham, M., 2013, Vulnerability Scrying Method For Software Vulnerability Discovery Prediction Without A Vulnerability Database, *Ieee Transactions On Reliability*, 62 (2), 395-407.

Rescorla, E., 2005, Is Finding Security Holes A Good Idea?, *Ieee Security & Privacy*, 3 (1), 14-19.

Sarkar, S., Goel, N. K. and Mathur, B. S., 2009, Adequacy Of Nakagami- M Distribution Function To Derive Giuh, *Journal Of Hydrologic Engineering*, 14 (10), 1070-1079.

Sarkar, S., Goel, N. K. and Mathur, B. S., 2010, Performance Investigation Of Nakagami- M Distribution To Derive Flood Hydrograph By Genetic Algorithm Optimization Approach, *Journal Of Hydrologic Engineering*, 15 (8), 658-666.

Scandariato, R. and Walden, J., 2012, Predicting Vulnerable Classes In An Android Application. Proceedings Of The 4th International Workshop On Security Measurements And Metrics. Lund, Sweden, Acm**:** 11-16.

Scandariato, R., Walden, J., Hovsepyan, A. and Joosen, W., 2014, Predicting Vulnerable Software Components Via Text Mining, *Ieee Transactions On Software Engineering*, 40 (10), 993-1006.

Shankar, P. M., Piccoli, C. W., Reid, J. M., Forsberg, F. and Goldberg, B. B., 2005, Application Of The Compound Probability Density Function For Characterization Of Breast Masses In Ultrasound B Scans, *Physics In Medicine And Biology*, 50 (10), 2241-2248.

Shoukri, M. M., Mian, I. U. H. and Tracy, D. S., 1988, Sampling Properties Of Estimators Of The Log-Logistic Distribution With Application To Canadian Precipitation Data, *Canadian Journal Of Statistics*, 16 (3), 223-236.

Smirnov, N., 1939, On The Estimation Of The Discrepancy Between Emprical Curves Of Distribution For Two Independent Samples, *Bulletin Mathématique De L'Université De Moscow*, 2 (2), 3-11.

Tsui, P.-H., Huang, C.-C. and Wang, S.-H., 2006, Use Of Nakagami Distribution And Logarithmic Compression In Ultrasonic Tissue Characterization, *Journal Of Medical And Biological Engineering*, 26 (2), 69.

Türksen, I. B., Khaniyev, T. and Gokpinar, F., 2015, Investigation Of Fuzzy Inventory Model Of Type (S, S) With Nakagami Distributed Demands, *Journal Of Intelligent & Fuzzy Systems*, 29 (2), 531-538.

Ucal, M. Ş., 2006, Ekonometrik Model Seçim Kriterleri Üzerine Kisa Bir İnceleme, *C.Ü. İktisadi Ve İdari Bilimler Fakültesi*, 7 (2), 41-57.

Wang, X., Ma, R., Li, B., Tian, D. and Wang, X., 2019, E-Wbm: An Effort-Based Vulnerability Discovery Model, *Ieee Access*, 7, 44276-44292.

Woo, S.-W., Alhazmi, O. and Malaiya, Y., 2006a, An Analysis Of The Vulnerability Discovery Process In Web Browsers. Proceeding Of The 10th Iasted International Conferance Software Engineering And Applicaitons. Usa**:** 172-177.

Woo, S.-W., Joh, H., Alhazmi, O. H. and Malaiya, Y. K., 2011, Modeling Vulnerability Discovery Process In Apache And Iis Http Servers, *Computers & Security*, 30 (1), 50-62.

Woo, S., Alhazmi, O. H. and Malaiya, Y. K., 2006b, Assessing Vulnerabilities In Apache And Iis Http Servers, *2006 2nd Ieee International Symposium On Dependable, Autonomic And Secure Computing*, 103-110.

Younis, A. A., Joh, H. and Malaiya, Y. K., 2011, Modeling Learningless Vulnerability Discovery Using A Folded Distribution, *The 2011 International Conference On Security And Management*, Usa, 1-10.