Using Historical Software Vulnerability Data to Forecast Future Vulnerabilities

David Last
Air Force Research Laboratory
Information Directorate, RISB
Rome, NY USA
david.last.1@us.af.mil

***Abstract*—The field of network and computer security is a never-ending race with attackers, trying to identify and patch software vulnerabilities before they can be exploited. In this ongoing conflict, it would be quite useful to be able to predict when and where the next software vulnerability would appear. The research presented in this paper is the first step towards a capability for forecasting vulnerability discovery rates for individual software packages. This first step involves creating forecast models for vulnerability rates at the global level, as well as the category (web browser, operating system, and video player) level. These models will later be used as a factor in the predictive models for individual software packages. A number of regression models are fit to historical vulnerability data from the National Vulnerability Database (NVD) to identify historical trends in vulnerability discovery. Then, k-NN classification is used in conjunction with several time series distance measurements to select the appropriate regression models for a forecast. 68% and 95% confidence bounds are generated around the actual forecast to provide a margin of error. Experimentation using this method on the NVD data demonstrates the accuracy of these forecasts, as well as the accuracy of the confidence bounds forecasts. Analysis of these results indicates which time series distance measures produce the best vulnerability discovery forecasts.**

Keywords—cybersecurity, vulnerability prediction, vulnerability discovery model

I. INTRODUCTION

It seems that almost every week we hear in the news about some new critical software vulnerability, or an attack that was successfully carried out exploiting one of these vulnerabilities. Well-publicized vulnerabilities such as the Heartbleed bug, the Shellshock bug, and the vulnerabilities exploited by Stuxnet constantly remind us of the ongoing struggle between cyber attackers and defenders. Due to the increasing number of vulnerabilities in critical information systems and also due to limited IT budgets, cyber defenders must carefully allocate their limited defense resources to protect the weak areas in their networks. This allocation can be based on the concentration of reported software vulnerabilities; however, network defenders also need to protect systems against as-yet-unreported vulnerabilities. The problem of forecasting future software vulnerabilities has been previously addressed by vulnerability discovery models [1] [2] [3]; most of this research has applied vulnerability forecasting techniques to a few select software packages. This paper is the first installment of research geared towards extending generalizable vulnerability forecast models to all software packages.

This research will produce vulnerability forecast models for specific software packages (e.g., Internet Explorer). These models will incorporate a number of variables affecting vulnerability discovery, such as code function, code complexity, code length, historical quality of code being produced by a certain vendor, enthusiasm of the vulnerability hunters, etc. Vulnerability hunter enthusiasm that affects vulnerability discovery rates for Internet Explorer will include enthusiasm at a global level, enthusiasm for the specific category (web browsers), and enthusiasm for Internet Explorer itself. The research presented in this paper will model global and category enthusiasm based on vulnerability discovery forecasts at the global and category level. In this research, we will utilize a number of regression models and then use *k*-NN classification to select the best regression model for forecast based on the historical performance of the regression models for the particular dataset at the particular time. Several well-established time series-matching distance measures will be used in conjunction with the *k*-NN classification. In addition, the analysis will generate confidence bounds corresponding to the confidence in the forecast. In later research, these global and category vulnerability forecast models will be used to model vulnerability hunter enthusiasm and will be incorporated into vulnerability forecast models for specific software packages such as Internet Explorer. This in turn will inform network defenders when deciding the amount of limited defense resources to deploy in defense of Internet Explorer installations.

The current state of the research in this area will be summarized in the following sections. Section II will discuss previous research efforts in this area. Section III will detail how this work has been expanded in the current research. Section IV will examine the effectiveness of the forecast models developed in this research when applied to real-world data, and Section V will discuss the planned continuation of this research and refinement of the forecast models.

II. RELATED WORK

Historically, different researchers have investigated this topic from various angles. Some of the more prolific researchers in this

978-1-4799-8594-4/15/$31.00 ©2015 IEEE

field are Alhazmi and Malaiya. They, along with others, have adapted Software Reliability Growth Models (SRGMs) [4] into Vulnerability Discovery Models (VDMs), which model the rate at which vulnerabilities will be discovered for a particular software package over its lifetime [1]. They apply different VDMs to historical software vulnerability data for major operating systems [5] [6] as well as examining the predictive abilities of these VDMs against a small number of operating systems [7]. They have also adapted VDMs to explain vulnerability discovery in multi-version software systems [2]. Their work forms a sound scientific basis for vulnerability prediction, but they do not investigate the forecast accuracy of their methods over a broad spectrum of software packages. Joh and Malaiya [8] also performed an investigation into the seasonality of vulnerability reporting, to see if different software packages had more reported vulnerabilities during certain months of the year. Their results were intriguing, but they did not leverage their findings into a forecasting capability; their findings could be combined with the research presented in this paper to form a more complete picture of vulnerability hunter enthusiasm.

Other researchers have also investigated software vulnerability discovery forecasting. Zhang, Caragea, and Ou [3] attempted to predict the Time To Next Vulnerability (TTNV) for specific software packages based on vulnerability discovery trends in the NVD. Their focus on TTNV makes their research more narrowly focused than the research presented in this paper.

In addition to vulnerability prediction based on trends in historical data, other research has been conducted that does not rely on large amounts of historical data. Rahimi and Zargham [9] theorized that a particular software package's vulnerability discovery rate would be directly correlated to the quality of the code and inversely correlated to the code's complexity. They sought to use this theory to predict the rate of vulnerability discovery for a particular software package without the benefit of historical data. Their methods delivered a high level of success when used to predict the vulnerability discovery rate for three particular software packages for which the ground truth vulnerability discovery rate was known. However, their method depends on source code analysis, and its usefulness is thus limited to open source software.

III. Vulnerability Discovery Forecast Based on Historical Data

A. Data

This research uses vulnerability reporting data from the National Vulnerability Database (NVD) [10]. The National Institute of Science and Technology (NIST) maintains a public database of all reported software vulnerabilities, with comprehensive reporting beginning around 2001. Each vulnerability in the NVD is scored according to the Common Vulnerability Scoring System (CVSS). The CVSS score for a vulnerability is based on six characteristics: Access Complexity, Authentication, Access Vector, Confidentiality Impact, Integrity Impact, and Availability Impact. These six characteristics are used to calculate CVSS Impact, Exploitability, and Base Scores, which give an idea of the severity of the vulnerability.

Although the NVD records are not exhaustive and a small number of the records contain errors [3], this compilation is the most complete and best-suited for this research among all available vulnerability databases. One drawback to NVD and similar databases is that it records the date a vulnerability was *reported*, not the date it was *discovered*. This will affect the fidelity of vulnerability discovery trend analysis. For the purposes of this research, the NVD dataset is divided into several subsets. The GlobalVulns dataset contains all vulnerabilities in the NVD. The BrowserVulns, OSVulns, and VideoVulns datasets contain vulnerabilities affecting the software packages listed in Table 1.

TABLE I. SOFTWARE PACKAGES INCLUDED IN NVD ANALYSIS.

BrowserVulns	OSVulns	VideoVulns
Microsoft Internet Explorer	Microsoft Windows	RealNetworks RealPlayer
Mozilla Firefox	Apple OS X	Adobe FlashPlayer
Google Chrome	Ubuntu Linux	FFmpeg
Apple Safari	Debian Linux	Adobe ShockWave
Opera	FreeBSD	Apple QuickTime
Mozilla Seamonkey	HP HP-UX	Windows MediaPlayer
	IBM AIX	VideoLAN VLC Media Player
	Linux Kernel	
	NetBSD Linux	
	Redhat Linux kernel	
	Sun Solaris / OpenSolaris	
	Novell Suse / OpenSuse Linux	

B. Regression Models

The analysis in this research examines the lifetime cumulative vulnerabilities discovered up to a given month rather than vulnerabilities discovered per month. This is a well-established approach in this research area [7] [2]; it focuses on overall trends rather than month-to-month performance. This research utilizes a number of regression models in order to generate forecasts. For each of the regression models, a line is fit to historical data in the *training period* according to the specifics of the model; that line is then extended into the future to generate the vulnerability discovery forecast during the *testing* (or *forecast*) *period*.

This research uses three basic regression models. The *linear regression* model is based on the assumption of (relatively) unchanging vulnerability discovery rates over the length of the forecast, and the *quadratic regression* model assumes increasing or decreasing vulnerability discovery rates. The *combined regression* model is a risk-averse model that averages the linear and quadratic forecasts together.

These three models can identify a single trend across the training period; however, both long- and short-term trends may affect the vulnerability discovery rate during the *forecast period*. The three basic regression models can be used in one of four variations based on the incorporation of long- and short-term trends. The *simple time horizon* variation utilizes a single linear, quadratic, or combined regression model. The *disjoint time horizon* uses regression models fit to short-, mid-, and long-term *training period* data (e.g., 25%, 50%, and 100% of the available historical data, respectively) to forecast short-, mid-, and long-term trends. However, this approach ignores the effect long-term trends may have on the short-term forecast, etc. This deficiency is addressed by the *homogeneous time horizon* variation; in this

variation, the forecasts based on the short-, mid-, and long-term training period curves are averaged together to generate a single (long-term) forecast. The *proportional time horizon* variation is based on the realization that long-term trends will likely have a greater impact on the long-term forecast than will short-term trends, etc. In this variation, the forecasts based on the short-, mid-, and long-term regression models are weighted according to their importance to the different time horizons of the forecast, and averaged together. Given the three basic models and the four variations, there are twelve distinct regression models that will be evaluated for this research.

We may use one of two measures to evaluate the accuracy of a forecast compared to the true data. The *Mean Absolute Percent Error* (*MAPE*) measures the deviation between the forecast and the true data; the *Root Mean Square Percent Error* (*RMSPE*) measures the same, but assigns a higher penalty to outliers.

C. Generating the Forecast

1) Selecting the Best Regression Model for Forecast

For each of the twelve regression model variations, there are multiple choices of training period length; each training period length may identify different trends and generate different forecasts. Each unique combination of regression model variation and training period length is termed a *Combined Model*. The basic question of this research is what is the best Combined Model to generate a forecast on a particular dataset at a particular time? This question is similar to the main question addressed by the study of time series matching [11]. This research area examines the process of searching large datasets for time series that match a given query element under a given similarity criteria. The most common approach to this problem is *k-Nearest Neighbor* (*k*-NN) classification [11]. In order to use *k*-NN classification for time series matching, a number of known time series called *training samples* are previously assigned to different classes (for example, network bandwidth usage profiles from different times of the day could be assigned to classes associated with different time periods over the day). An unclassified sample is compared to the training samples by measuring the "distance" between the unclassified sample and the training samples using some time series matching distance measure (discussed in a following paragraph). The unclassified sample is assigned to the class that has the most representatives in the set of *k* samples closest to unclassified sample. In a *weighted k*-NN classification, the training samples may be assigned weights which affect their contribution to the classification decision.

The decision regarding which Combined Model to use for a forecast on a given dataset at a particular point in time is accomplished using *k*-NN classification. The historical data in the dataset is divided into a number of overlapping train/validate periods (Fig. 1). For each train/validate period, each Combined Model is fit to the training data, and a forecast is generated over the validation data. Using one of the time series matching distance measures discussed below, the distance between the forecast and the true data is calculated. The Combined Model's forecast for the particular train/validate period becomes one of the training samples; the distance measured becomes the distance between the training sample and the unclassified sample representing the desired forecast (Fig. 1). For the *k*-NN classification, *k* is set equal to the total number of training samples generated, and each training sample is assigned a distance weight equal to its distance from the unclassified sample. Each training sample is also assigned a temporal weight according to the age of the validate period, with more recent validate periods weighted more heavily (Equation 1). Therefore, the aggregate weight for each training sample is its distance weight multiplied by its temporal weight. The class (Combined Model) whose training samples have the lowest sum aggregate weights is selected as the class for the unclassified sample (i.e., this Combined Model is used to generate the desired forecast).

Since more recent trends in the dataset are likely to have a

Fig. 1. Selecting the best Combined Model to generate a forecast.

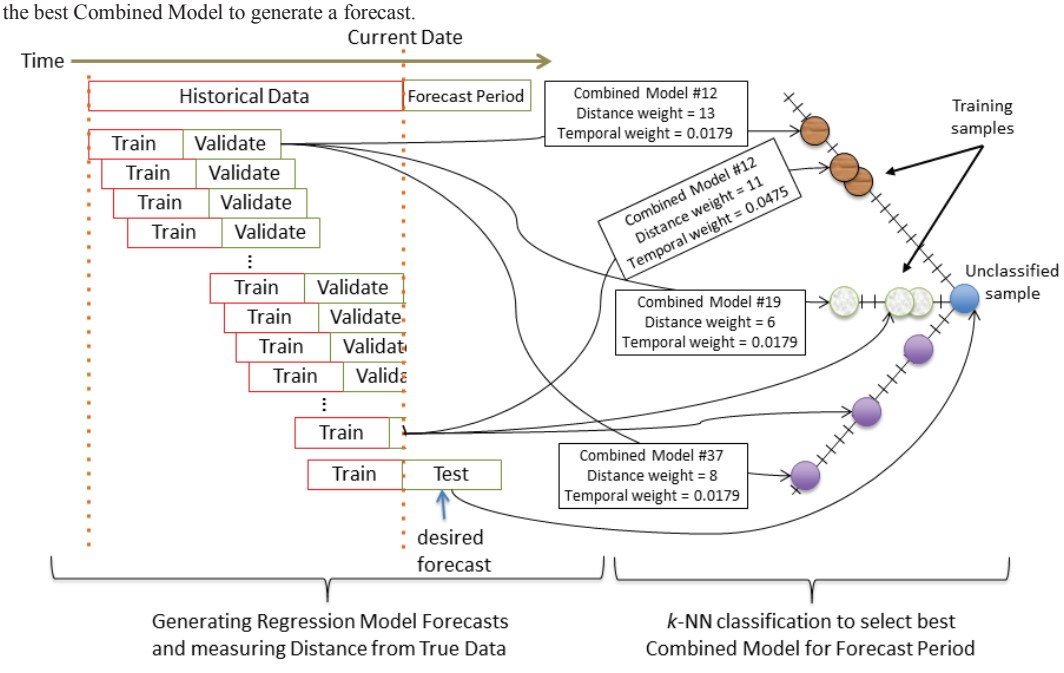


Fig. 2. Standard Deviation confidence bounds.

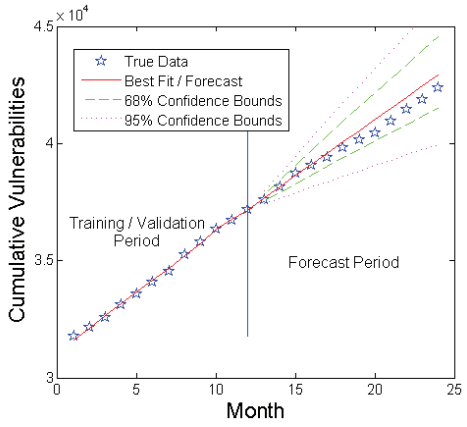


Fig. 3. Percentile confidence bounds.

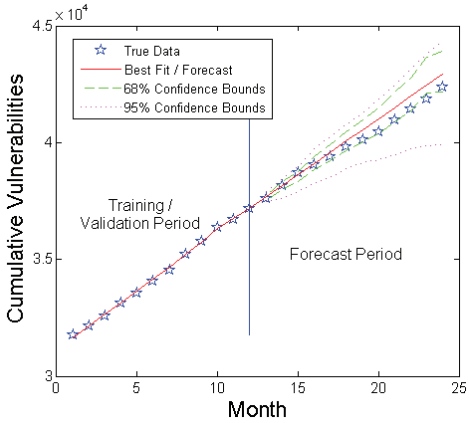
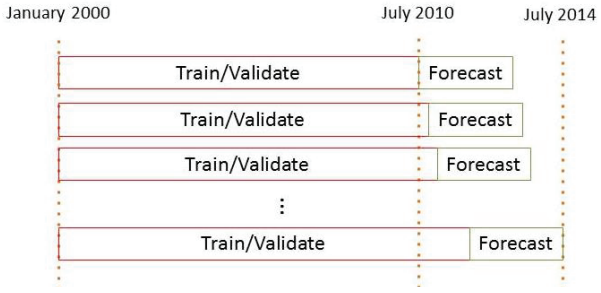


Fig. 4. Train, validate, and forecast data for experimental setup.



greater impact on the true data in the forecast period, it is desirable to more heavily weight the contributions or more recent training samples. This also holds when generating Confidence Bounds (discussed below). This research uses a standard temporal weighting scheme using the weighting factor ρ in the following equation.

$$temporal\_weight_i = (1-\rho)\rho^i, \quad i = t_{current} - t_{datapoint\,i}, \quad 0 < \rho < 1 \quad (1)$$

In the field of time-series matching, there are a number of different distance measures that are used to calculate the distance between two time series. Several of these measures are used to measure the distance between a validation forecast and the true data in Fig. 1. The simplest class of distance measures are non-elastic and do not support time shifting. The Lp-norms are a common example of this type of distance measure, with the *Euclidean Distance* (*ED*) being the L2-norm [11]. The second class contains elastic distance measures that do tolerate time shifting, but are not true metrics according to the triangle inequality; *Dynamic Time Warping* (*DTW*) falls into this class [12]. The third class consists of elastic metrics that tolerate time shifting; *Edit Distance on Real Sequences* (*EDR*) [13] and *Time Warp Edit Distance* (*TWED*) [14] fall into this category. In this research, MAPE and RMSPE will also be evaluated for their effectiveness as distance measures.

2) Confidence Bounds

Given that no forecast will be perfect, it is also useful to measure the margin of error in the forecast. In this research, we introduce the idea of the "*X*% confidence bound"; given a forecast, we give an *X*% probability that any given month's true datapoint during the forecast period will fall within the confidence bounds. When dealing with normal distributions, the standard deviation is used to generate 68% and 95% confidence bounds; therefore, we will deal with 68% and 95% confidence bounds as a matter of convention.

In this research, we will use two methods for generating confidence bounds. For the *Standard Deviation Method*, we will treat the month-to-month percent errors of the train/validate forecasts (*trainging samples* in Fig. 1) as a normal distribution. The standard deviation of the percent errors are used to generate 68% and 95% confidence bounds in the normal way (Fig. 2). However, analysis indicates that these percent errors do not often follow a normal distribution, so a second method is needed. For the *Percentile Method*, we can translate the 68% confidence bounds into a 68th percentile. We examine all of the 68 percentile ranges in the validation (*trainging samples* in Fig. 1) percent error dataset (i.e., 1st-69th percentile, 2nd-70th percentile, etc.). Depending on the analysis parameter, either the narrowest, widest, or median-width percentile range is used to generate the confidence bounds (Fig. 3). For this research, unique confidence bounds are generated for each month of the forecast (1st month, 2nd month, etc.) based on the percent errors from the same month in the validation forecasts. This is done to account for any trends that may be unique to certain months of the forecast.

IV. Experimentation

A. Experimental setup

The experiments for this research were conducted on the GlobalVulns, BrowserVulns, OSVulns, and VideoVulns datasets. Data spanning from January 2000 to July 2011 was designated as train/validate data, and the time period from July 2011 to April 2015 were designated as the forecast period. For each experiment, forecasts were made beginning on each month in the forecast period, with the entire dataset preceding that month as the train/validate period (Fig. 4). This series of forecasts simulates a network defender using this vulnerability forecasting method to generate new forecasts every month; the accuracy of these forecasts over time may then be analyzed. Training periods of 12, 24, 36, and 48 months were used, and validation/test/forecast periods of 12, 18, and 24 months were used. With 12 regression model variations and 4 training period lengths, a total of 48 Combined Models were trained over the course of the experiment. The TWED, EDR, and DTW distance measures involve

Fig. 5. Accuracy of forecasts (as measured by median RMSPE) for GlobalVulns dataset for 12-, 18-, and 24-month forecasts.

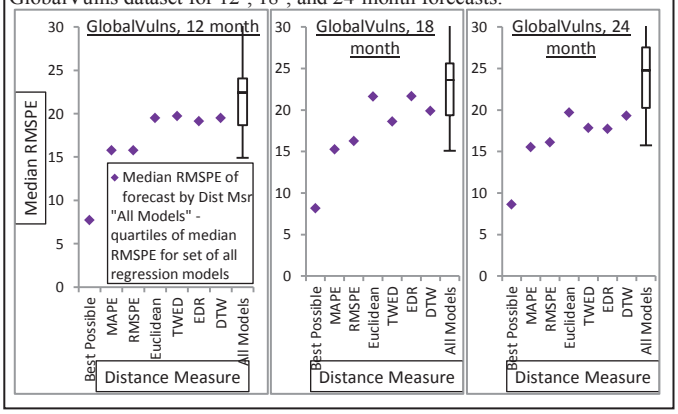


Fig. 6. Accuracy of forecasts (as measured by median RMSPE) for BrowserVulns dataset for 12-, 18-, and 24-month forecasts.

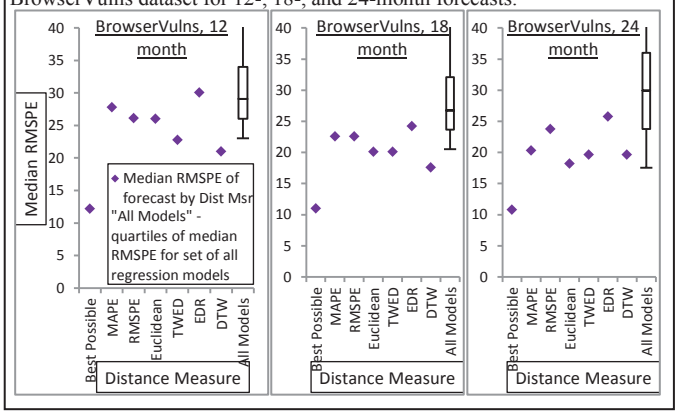


Fig. 7. Accuracy of forecasts (as measured by median RMSPE) for OSVulns dataset for 12-, 18-, and 24-month forecasts.

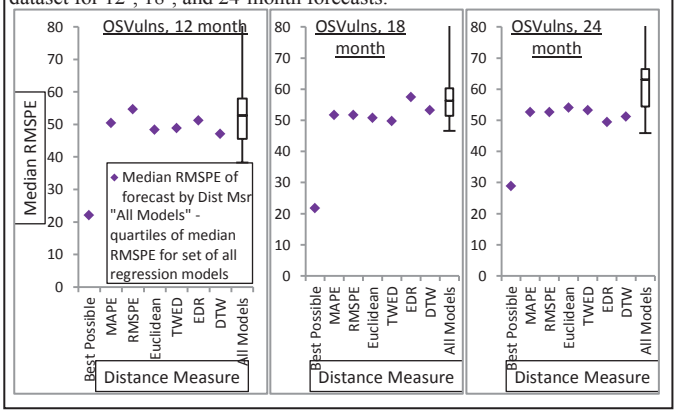
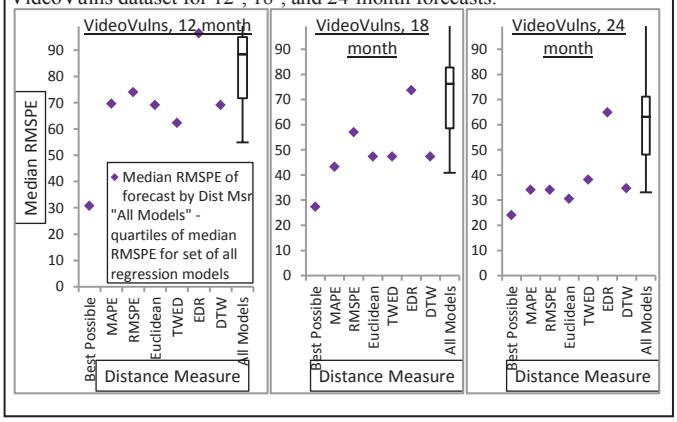


Fig. 8. Accuracy of forecasts (as measured by median RMSPE) for VideoVulns dataset for 12-, 18-, and 24-month forecasts.



configurable parameters; the values used for these parameters were chosen as the median value of the recommended ranges as defined in [11]. For TWED, λ was set to 0.5 and ν was set to 0.001. For EDR, ε was set to 0.5σ. For DTW, ω was set to $0.125N$. For all experiments reported here, the percentile confidence bounds method using the widest percentile was used, and ρ was set to 0.95 for temporal weighting.

B. Experiment 1: Forecast Accuracy

The first experiment was designed to answer the question, "How well does the distance measure classification generate accurate forecasts (as measured by RMSPE)?" This question was answered using measurements over each forecast period in Fig. 4. For each forecast period, the Combined Model whose forecast resulted in the lowest RMSPE was selected, and the RMSPE for each forecast period's best-fit Combined Model were collected. This formed the *Best Possible* baseline, or the highest possible accuracy forecasts if the classification had chosen the most accurate (lowest RMSPE) Combined Model each time. Next, for each forecast period, the Combined Model that was selected according to the classification using MAPE, RMSPE, ED, TWED, EDR, or DTW was chosen, and the RMPSE of the forecasts from the selected Combined Models were collected across each forecast period. Finally, the RMSPE from exclusive use of each Combined Model were collected. The purpose of these measurements was to determine if the forecasts resulting from the distance measure classifications had lower median RMSPE than most (or all) of the single Combined Model forecasts.

The results of this experiment are reported in Fig. 5 through Fig. 8. These plots show the median RMSPE for the *Best Fit* baseline and the forecasts generated by classification using different distance measures, as well as the 1st-4th quartiles of median RMSPE for single-Combined Model forecasts.

The experimental results demonstrate that no distance measure classification method can generate forecasts close to the Best Possible baseline. Additionally, the Best Possible baseline and the classification forecasts display worse RMSPE across successive datasets. Both of these phenomena are attributable to the same cause. In order for the forecast models to generate accurate predictions, the underlying datasets must be smooth (i.e., display consistent trends rather than sudden jumps). Manual inspection shows that each successive dataset (GlobalVulns, BrowserVulns, OSVulns, and VideoVulns) is less smooth. This experiment demonstrates an intuitive truth: the success of forecast models is limited by the smoothness of the data. Beyond this, however, classification by MAPE and RMSPE generate the most accurate forecasts relative to the single Combined Model forecasts for the GlobalVulns dataset (Fig. 5), while Euclidean, DTW, and TWED generate the most accurate forecasts for the BrowserVulns, OSVulns, and VideoVulns datasets (Fig. 6-8). EDR on average provides the least accurate forecasts across all datasets.

Fig. 9. Percentage of true testing data points falling within the confidence bound generated by distance measures for GlobalVulns dataset.

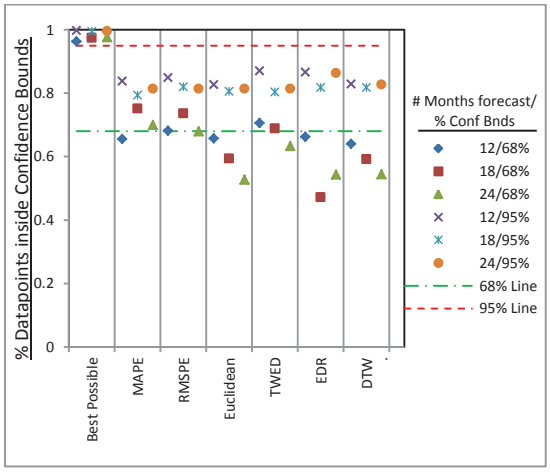


Fig. 10. Percentage of true testing data points falling within the confidence bound generated by distance measures for BrowserVulns dataset.

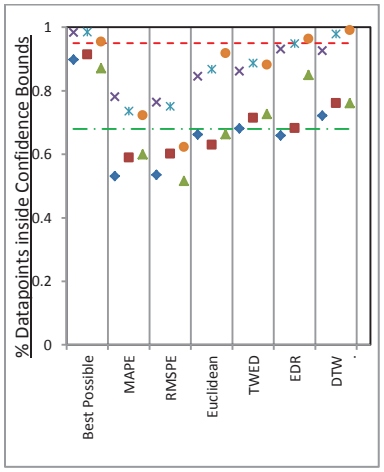


Fig. 11. Percentage of true testing data points falling within the confidence bound generated by distance measures for OSVulns dataset.

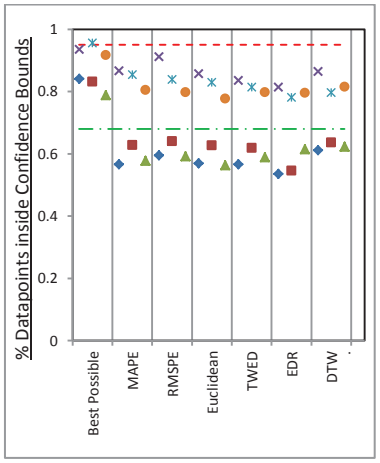
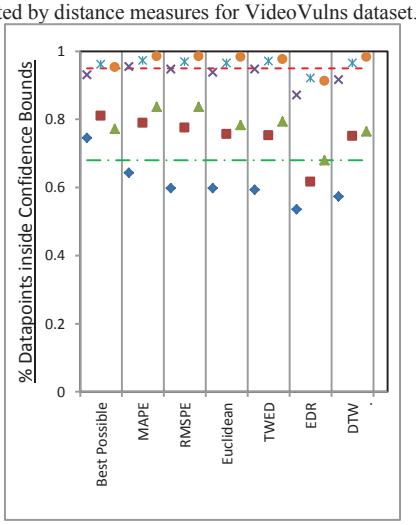


Fig. 12. Percentage of true testing data points falling within the confidence bound generated by distance measures for VideoVulns dataset.



For the GlobalVulns and OSVulns datasets, median RMSPE generated by distance measure classification forecasts remained relatively consistent across the 12-, 18-, and 24-month forecasts. However, for the BrowserVulns and VideoVulns datasets, the median RMSPE trended downwards as the forecasts grew longer. These results support the theory that there are long-term trends present in the data that are identifiable in the short forecasts for smoother datasets (GlobalVulns and OSVulns), and that these long-term trends are identifiable but somewhat masked by short-term volatility in less smooth datasets (BrowserVulns and VideoVulns). One possible explanation for this long-term trend/short-term volatility could be the seasonality present in vulnerability reports that was explored in [8].

C. *Experiment 2: Confidence Bounds Accuracy*

In the introduction of this paper, it was suggested that one of the main beneficiaries of this research would be a network defender who wishes to identify the areas of the network likely to display new vulnerabilities over the next 12-24 months. This information would allow such a defender to deploy available defense resources in an efficient manner. From this perspective, the forecast is of some interest; however, the main focus should be on the confidence bounds. Since the confidence bounds define the best- and worst-case scenarios, this will help the network defender determine what defenses should be deployed to defend against likely scenarios of vulnerability discovery rates. The second experiment in this research was designed to answer the question, "How well does the distance measure classification generate accurate confidence bounds, as measured by the percentage of datapoints falling within the confidence bounds?" For this experiment, 68% and 95% confidence bounds were generated for each forecast period in Fig. 4, using the percentile method with the widest percentile range and $\rho = 0.95$. Confidence bounds were generated based on the Best Possible Combined Model from the first experiment, as well as from the forecast models selected by the classifications by different distance measures. The percentage of datapoints falling within the 68% and 95% confidence bounds were tabulated, and the results are shown in Fig. 9 through Fig. 12.

At first glance, it is easy to see that the majority of the confidence bounds exceed or come close to meeting their 68% or 95% targets. However, the confidence bounds perform poorly for the GlobalVulns dataset relative to the other datasets. This could be the result of confidence bounds generated from "smoother" data being less able to account for outliers (95% confidence bounds). Beyond that, the best distance measures for generating forecasts as noted from the previous experiment generally perform well in generating confidence bounds as well. For the BrowserVulns and VideoVulns datasets, the confidence bounds generally perform better over longer forecast periods. This phenomenon could be indicative of a strong influence of the seasonality in vulnerability discovery discussed in [8], resulting in long-term trends masked by short-term volatility.

V. Future Work

There are several avenues for future work in this research. The research presented here attempts to predict the cumulative number of future software vulnerabilities; this information will be used to model vulnerability hunter enthusiasm. However, this analysis treats all vulnerabilities equally, whether they were easier

or more difficult to discover. In order to more accurately model enthusiasm, this difference must be taken into account. The NVD assigns a Common Vulnerability Severity Score (CVSS) Access Complexity Score to each vulnerability. This metric measures the complexity of the attack required to exploit the vulnerability once the attacker has gained access to the target system; this metric may serve as an analogue for how difficult it may have been to find. Repeating the analysis from this paper while weighting vulnerabilities according to their Access Complexity Score may provide a better model of enthusiasm with better forecasting capabilities.

The temporal weighting of data discussed in this paper assumes that distance in time between historical data and the forecast period is the only criterion by which the first influences the second. However, some of the experimental results presented in this paper suggest that the seasonality discussed in [8] may also influence vulnerability discovery trends. In future work, an alternate weighting scheme that considers the calendar month of historical data will be used to improve forecasting capabilities.

Ultimately, the outputs of this research will be incorporated into software package-specific vulnerability discovery models. These models will enable a network defender to make informed decisions on defense deployment based on expected future vulnerability of the software installed on the network.

VI. Conclusion

The ability to predict where and when software vulnerabilities will appear would be useful in many contexts, not the least of which would be for a network defender determining where to deploy available network defenses. Much research has been conducted in pursuit of such a predictive ability. The research presented in this paper seeks to leverage trends in historical vulnerability discovery data in generating forecasts for the discovery of new vulnerabilities. This research utilizes the established machine learning method of *k*-NN classification in pursuit of this predictive capability. One of the lessons to be drawn from this research is that the quality of vulnerability discovery forecasts can only be as good as the prevalence of modeled trends (i.e., "smoothness") in the data. This realization should temper the expectations set for these forecasts. Still, the classification techniques developed here can achieve better forecasts than most single Combined Models. Classification of models based on the Mean Absolute Percent Error (*MAPE*) and Root Mean Square Percent Error (*RMSPE*) distance measures tend to generate better forecasts over smoother data, while classification based on the Euclidean Distance (*ED*), Time Warp Edit Distance (*TWED*), and Dynamic Time Warping (*DTW*) perform well for less smooth data. The choice of which distance measure should be used will be determined by the particular desired criteria set for the forecasts. This work will be continued in pursuit of the development of vulnerability prediction models for individual software packages.

References

[1] O. H. Alhazmi and Y. K. Malaiya, "Modeling the Vulnerability Discovery Process," in *The 16th IEEE International Symposium on Software Reliability Engineering (ISSRE'05)*, Washington, DC, USA, 2005.

[2] J. Kim, Y. K. Malaiya and I. Ray, "Vulnerability Discovery in Multi-Version Software Systems," in *10th IEEE High Assurance Systems Engineering Symposium 2007 (HASE'07)*, 2007.

[3] S. Zhang, D. Caragea and X. Ou, "An Empirical Study on Using the National Vulnerability Database to Predict Software Vulnerabilities," *Database and Expert Systems Applications,* pp. 217-231, 2011.

[4] Y. K. Malaiya and J. Denton, "What Do the Software Reliability Growth Model Parameters Represnt?," in *International Symposium on Software Reliability Engineering*, 1997.

[5] O. H. Alhazmi and Y. K. Malaiya, "Quantitative Vulnerability Assessment of Systems Software," in *Proceedings of the Annual Reliability and Maintainability Symposium*, 2005.

[6] O. H. Alhazmi and Y. K. Malaiya, "Application of Vulnerability Discovery Models to Major Operating Systems," *IEEE Transactions on Reliability,* vol. 57, no. 1, March 2008.

[7] O. H. Alhazmi and Y. K. Malaiya, "Prediction Capabilities of Vulnerability Discovery Models," in *The Annual Reliability and Maintainability Symposium 2006 (RAMS'06)*, 2006.

[8] H. Joh and Y. K. Malaiya, "Seasonal Variation in the Vulnerability Discovery Process," in *2009 International Conference on Software Testing Verification and Validation*, 2009.

[9] S. Rahimi and M. Zargham, "Vulnerability Scrying Method for Software Vulnerability Discovery Prediction Without a Vulnerability Database," *IEEE Transactions on Reliability,* vol. 62, no. 2, June 2013.

[10] "National Vulnerability Database," National Institute of Standards and Technology (NIST), [Online]. Available: https://nvd.nist.gov/. [Accessed 17 March 2015].

[11] J. Serra and j. L. Arcos, "An Empirical Evaluation of Similarity Measures for Time Series Classification," *Knowledge-Based Systems,* vol. 67, pp. 305-314, 2014.

[12] H. Sakoe and S. Chiba, "Dynamic Programming Algorithm Optimization for Spoken Word Recognition," *Acoustics, Speech and Signal Processing, IEEE Transactions on,* vol. 26, no. 1, pp. 43-49, 1978.

[13] L. Chen, M.T.Ozsu and V. Oria, "Robust and Fast Similarity Search for Moving Object Trajectories," in *Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data*, 2005.

[14] P. Marteau, "Time Warp Edit Distance with Stiffness Adjustment for Time Series Matching," *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* vol. 31, no. 2, pp. 306-318, 2009.