Understanding the Hidden Cost of Software Vulnerabilities: Measurements and Predictions

Afsah Anwar[1], Aminollah Khormali[2] and DaeHun Nyang[3], and Aziz Mohaisen[4]

University of Central Florida, Orlando, FL 32816, USA
[1]afsahanwar@Knights.ucf.edu,[2]aminkhormali@knights.ucf.edu,[4]mohaisen@ucf.edu
[3] Inha University, Incheon, Republic of Korea
nyang@inha.ac.kr

Abstract. Vulnerabilities have a detrimental effect on end-users and enterprises, both direct and indirect; including loss of private data, intellectual property, the competitive edge, performance, etc. Despite the growing software industry and a push towards a digital economy, enterprises are increasingly considering security as an added cost, which makes it necessary for those enterprises to see a tangible incentive in adopting security. Furthermore, despite data breach laws that are in place, prior studies have suggested that only 4% of reported data breach incidents have resulted in litigation in federal courts, showing the limited legal ramifications of security breaches and vulnerabilities.
In this paper, we study the hidden cost of software vulnerabilities reported in the National Vulnerability Database (NVD) through stock price analysis. Towards this goal, we perform a high-fidelity data augmentation to ensure data reliability and to estimate vulnerability disclosure dates as a baseline for estimating the implication of software vulnerabilities. We further build a model for stock price prediction using the NARX Neural Network model to estimate the effect of vulnerability disclosure on the stock price. Compared to prior work, which relies on linear regression models, our approach is shown to provide better accuracy. Our analysis also shows that the effect of vulnerabilities on vendors varies, and greatly depends on the specific software industry. Whereas some industries are shown statistically to be affected negatively by the release of software vulnerabilities, even when those vulnerabilities are not broadly covered by the media, some others were not affected at all.

Keywords: Vulnerability Economics; Prediction; National Vulnerability Database

1 Introduction

An ideal software should be defect-free, reliable and resilient. However, vulnerabilities are defects in software products, which expose the product and users to risk alike, for e.g.,, Distributed Denial of Service attacks [1,2] or typosquatting attacks [3]. When such defects happen, users prefer vendors who take such defects as a priority, fix them, report them to their users, and keep the community

as a whole immune to adversaries. Failure to do so would put vulnerable vendors at risk, whereby users seek different vendors, causing great losses.

In practice, vulnerabilities have multiple costs associated with them. For example, a vulnerability leads to loss of trust by users, tarnished brand reputation, and ultimately results in the loss of customer-base. To deal with vulnerabilities, vendors also incur additional costs in the form of developer-hours spent fixing them and redeploying fixes. As such, vulnerabilities could be a direct cause of losing a competitive edge in the global market to vendors less prone to them. For example, a study by the National Institute of Standards and Technology (NIST) estimated that the US economy looses about $60 Billion USD every year for patches development and redistribution, systems re-deployment, as well as direct productivity loss due to vulnerabilities [4].

To make matters worse, the number of security incidents and vulnerabilities have been growing exponentially, leading to a similar growth in resources required for fixing them. In 2012, for example, Knight Capital, a financial services company, lost $400 Million USD because of a bug in their code; the company bought shares at the *ask price* and sold them at the *bid price* [5]. Losses from WannaCry (2017), a ransomware attack in over 150 countries affecting more than 100,000 groups, is estimated to be $4 Billion USD [6]. Virus attacks, such as Love Bug (2000), SirCam (2001), Nimda (2001), and CodeRed (2001), have had an impact of $8.75 Billion, $1.25 Billion, $1.5 Billion and $2.75 Billion USD, respectively [7]. With deployment of software in critical infrastructure, vulnerabilities could have overwhelming impact. For example defects like the loss of radio contact between the air traffic controller and the pilots due to unexpected shutdown of voice communication system and crash of the backup system within a minute of it turning on, could cost lives [8].

The cost of vulnerabilities is a variable that does not depend only on the type of the vulnerability, but also the industry, potential users, and the severity of the vulnerability as seen by those users. For example, users of security or financial software are more likely to lose faith in their product, compared to general e-commerce applications. A more severe vulnerability is also more likely to impact a vendor than a minor software glitch. For example, a vulnerability that can be used to repeatedly launch a Denial of Service (DoS) attack could be viewed more severely by users than, say, an access control misconfiguration (e.g., 1-time access-token exposure).

For publicly-traded drug and auto vendors, Jarrell and Peltzman [9] demonstrated that recalling products has a detrimental impact on shareholder value. Conversely, though, researches have shown that software vendors may, on the one hand, not suffer any significant losses due to vulnerabilities [10], or, on the other hand, grow in profit and offerings despite the parallel growth in software vulnerabilities. However, there are also underlying costs associated with each software vulnerability, as mentioned above, and those costs are maybe invisible [10]. For example, Romanosky *et al.* [11] studied software-related data breaches in the United States, and found that 4% of them resulted in litigation

in federal courts, out of which 50% (2% of the original studied cases) won by the plaintiffs.

Contributions. In this paper, we quantitatively analyze the loss faced by software vendors due to software vulnerabilities, through the lenses of stock price and valuation. To this end, this work has the following contributions. (i) An evaluation of vulnerabilities, disclosed in the year 2016, from the National Vulnerability Database (NVD) and their impact on their vendors. (ii) An accurate method for predicting stock price of the next day using NARX Neural Network. (iii) Industry-impact correlation analysis, demonstrating that some industries are more prone to stock loss due to vulnerabilities than others. (iv) Vulnerability type analysis, indicating that different types have different powers of affecting the stock price of a vendor.

Our work stands out in the following aspects, compared to the prior work (more in section 2). First, unlike the prior work, which is event-based (tracks vulnerabilities that are only reported in the press), we use a comprehensive dataset of disclosed vulnerabilities in the National Vulnerability Database (NVD). Per Spanos and Angelis [12], 81.1% of the prior work they surveyed were limited to security breaches, while we focus on all software vulnerabilities. Furthermore, per the same source, 32.4% of the prior work used Lexis/Nexis (database of popular newspapers in the United States) as their source, 24.3% used the Data Loss Archive and Database (data for privacy breach), 13.5% used CNET (technology website), and 13.5% used Factiva (global news database). In this study, we uniquely focus on using NVD. (ii) We design a model to accurately predict stock for the next day to precisely measure the effect of a vulnerability. Our approach outperforms state-of-the-art approach using linear regression (e.g., while our mean-squared error (MSE) using ANN is below 0.6, using linear regression results in MSE of 6.24). (iii) Unlike the prior work, we did not exclude any vendors, as we considered publicly-traded vendors on NYSE, NASDAQ, Frankfurt, Other OTC, Taiwan, and LSE. Spanos and Angelis [12] in their survey found that 83.8% of the surveyed work used vendors that traded in a US stock market, 13.5% used vendors from different countries and only 2.9% (1 out of 34 works) used firms traded in TYO (the leading stock exchange in Japan) [12].

Organization. The rest of the paper is organized as follows: In section 2, we re-visit the literature. In section 3, we present our approach to the problem. In section 4, we present our prediction model. In section 5, we evaluate the results. In section 6 we further comment on the statistical significance of our results, followed by discussion, limitations and future work in section 7. We conclude the paper in section 8.

2 Related Work

Our work is an amalgam of different fields, where we connect the vulnerabilities to economic affect on vendor. Perceptions often relate vulnerabilities to effect on the end user. Little has been said and done from the vendor's perspective.

Effect on Vendor's Stock. Hovav and D'Archy [10], and Telang *et al.* [13] analyzed, in event-based studies, vulnerabilities and their impact on vendors. While Hovav and D'Archy have shown that market shows no signs of significant negative reaction due to vulnerabilities, Telang *et al.* show that a vendor on average loses 0.6% of its stock value due to vulnerabilities. Goel *et al.* [14] pointed out that security breaches have an adverse impact of about 1% on the market value of a vendor. Campbell *et al.* [15] observed a significant negative market reaction to information security breaches involving unauthorized access to confidential data, but no significant reaction to non-confidential breaches. Cavusoglu *et al.* [16] show that the announcement of Internet security breaches has a negative impact on the market value of vendors.

Bose *et al.* [17] show that each phishing alert leads to a loss of market capitalization that is at least US$ 411 million for a firm.

Vulnerability Analysis. Li and Paxson [18] outlined a method to approximate public disclosure date by scrapping reference links in NVD, which we use in this study. Nguyen and Massaci [19] pointed out that the vulnerable versions data in NVD is unreliable. Christey and Martin [20] outlined caveats with the NVD data, also suggesting its unreliability. Romanosky *et al.* [21] found that data breach disclosure laws, on average, reduce identity theft caused by data breaches by 6.1%. Similarly, Gordon *et al.* [22] found a significant downward shift in impact post the September 11 attacks.

Financial Impact of Defects. Jarrell and Peltzman [9] analyzed the impact of recall in the drug and auto industries on vendors' stock value loss. Towards calculating the effect of a vulnerability, it is crucial to predict a hypothetical stock valuation in the absence of a vulnerability. Kar [23] suggested the use of Artificial Neural Network (ANN) as a reliable method for predicting stock value. Farhang *et al.* [24], suggest that higher security investments in Android devices do not impose higher product prices on customers.

3 Methodology

Using the information available on the National Vulnerability Database (NVD), the goal of this study is to track the public disclosure date of vulnerabilities and capture their impact on vendors stock market valuation. As in the prior work [9], we consider the fluctuation in the stock price as a measure of the reported vulnerabilities' impact. To this end, we calculate the impact on the following days, with respect to the predicted value of the stock on the day of vulnerability disclosure. However, we limit ourselves up to the third day of the public disclosure of the vulnerability to reduce the likelihood of interference with factors that might affect the market value. The rest of this section explains in details the steps taken to achieve the above goal.

3.1 Data and Data Augmentation

Our main sources of data are NVD [25] and Yahoo Finance [26]. Figure 1 summarizes, at a high-level, the flow of data creation, from the source of data to the

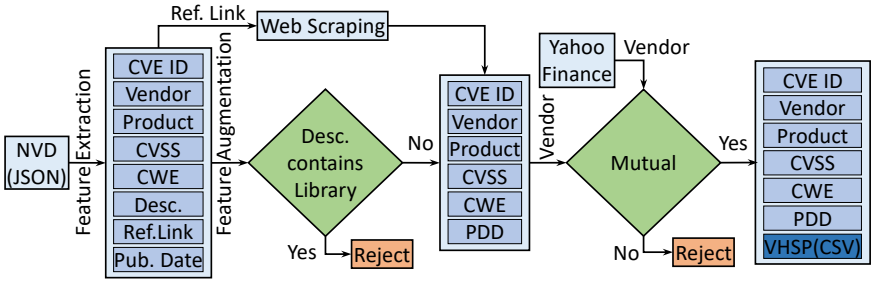

Fig. 1: Dataset Creation Flow. *Desc.* stands for the description of vulnerability, *Ref. Link* is the link referring to details corresponding to the vulnerability, *Pub. Date* is the Published Date, *CVSS* is Common Vulnerability Scoring System metrics, *CWE* is the Common Weakness Enumeration identifier, *PDD* is the Public Disclosure Date, approximated as the minimum of the dates gathered from the links corresponding to a vulnerability, and *VHSP* is the Vendor Historical Stock Price downloaded of mutual vendors from Yahoo Finance.

final dataset. In a nutshell, we extract information from JSON files downloaded from the National Vulnerability Database (NVD), scrape through the reference links for each vulnerability provided by NVD to approximate the disclosure date of the vulnerability, then check for indicative words, such as "lib" or "library" in the description of the vulnerability. If such words do not exist in the description, which means that those vulnerabilities are more likely associated with the vendor and not due to a third party, we consider the vulnerability for further analysis. We check for the vendor's historical stock prices using the Yahoo Finance. If the vendor exists in Yahoo Finance, we consider the vendor for our analysis, otherwise the vendor is rejected.

National Vulnerability Database (NVD) is a vulnerability database maintained by the National Institute of Standards and Technology (NIST) and contains all vulnerabilities reported to MITRE [27]. Analysts at NVD analyze the reported vulnerabilities, then insert them into the database after adding other necessary information, including (most importantly) a Common Vulnerabilities and Exposures Identifier (CVE-ID). In the following we elaborate on the other data elements in NVD associated with each vulnerability.

The NVD includes the following information (elements) for each reported vulnerability: the CVE-ID, vendor, product, Common Vulnerability Scoring System (CVSS) label, published date, Common Weakness Enumeration Identifier (CWE-ID) [28], description, and reference links. The CVSS label is provided using both version 2 and version 3 [29,30], which are widely used standard scoring techniques. The *vendor* element is the name of the vendor of the software that has the vulnerability, the *product* element is the name of the product which contains the vulnerability, and the *CVSS* is the severity of the vulnerability.

CVSS version 3, released in the later half of 2015, labels vulnerabilities as LOW, MEDIUM, HIGH, and CRITICAL, while the version 2 classifies them into LOW, MEDIUM, and HIGH. The attribute *published date* indicates the date when the vulnerability was entered into the NVD, while CWE-ID refers to the type of the weakness. The *description* element is a textual content to contextualize the submitted vulnerability. The *reference links* element is a set of the external URLs linking to references with additional details about the vulnerability, including a security advisory, a security thread, an email thread or a patch.

Data Preprocessing and Augmentation. The NVD data can be downloaded from the NVD website in either XML or JSON format; we chose the JSON format. The data is distributed in multiple JSON files with a file per year. We use the vulnerabilities reported in the year 2016, and limit our analysis to the severe ones. Since not all vulnerabilities have their CVSS version 3 assigned to them, we consider vulnerabilities with CVSS version 3 label as CRITICAL or version 2 label as "HIGH" to be severe. In our analysis we are interested in understanding the impact of core vulnerabilities in the software itself, rather than inherited vulnerabilities due to the use of third-party libraries. To this end, we filtered vulnerabilities due to third-party libraries by discarding those with the word "library" in their description. Given that a vulnerability may affect multiple vendors and products, we limit ourselves to the main source of the vulnerability by counting a vulnerability only under one vendor. For that, we checked the vendor name and the description in the vulnerability record, and found that the main vendor always appears in the description. Where multiple vendors appear in the description, we exclude those vulnerabilities from our analysis, since the vulnerability could be due to a third-party library common among products of those vendors. As a result, our dataset was reduced from 8,709 to 2,849 vulnerabilities.

Since the *published date* attribute captured in NVD is the date when the vulnerability was entered into the database and not the date when the vulnerability was actually found, the most important step in our analysis was to find the date when the vulnerability was disclosed to the public. We use the links present in the NVD to scrape through the web and label dates corresponding to each of the links, in an approach taken also by Li and Paxson [18]. We observed that some of the domains have stringent security measures preventing the automating scraping, while some did not have a date. For all such 1262 out of 8365 links, we manually visited the links and updated the corresponding URLs. For all URLs, we calculated the minimum of the dates corresponding to a vulnerability (when multiple dates are obtained from multiple URLs) and consider it as the public disclosure date. It should be noted that we ignore the links linking to patches, as the date of patching may or may not be same as the disclosure date, and market could only respond to public disclosure date.

In our dataset, we also found redundant vendor names, e.g., schneider-electric vs. schneider_electric, trendmicro vs. trend-micro, and palo_alto_networks vs. paloaltonetworks. We consolidate the various vendors under a consistent name, through manual inspection. For all the vendors in the above dataset we further

augment them by incorporating stock price over time from Yahoo Finance, as highlighted in the following.

Yahoo Finance For all the vulnerabilities in our dataset we gathered historical stock price information from Yahoo Finance. The historical data can be downloaded from Yahoo Finance as a Comma Separated Values (CSV) file. The file contains seven information attributes, namely, the date, open, low, high, close, adjusted Close, and volume. The *date* attribute corresponds to the date on which the stock's listed performance is captured. The *open* and *close* attributes are the stock value of the vendor on the given day at the opening and closing of the market, respectively. The *low* and *high* are the lowest and highest value of the vendor's stock achieved on the given day. The *adjusted close* attribute reflects the dividends and splits since that day. During an event of stock split, the adjusted closing price changes for every day in the history of the stock. For example, if stock for vendor X closed at $100 USD per share on December 5th, a 2:1 stock split is announced on December 6th, and the stock opened at $50 USD and closed at $60 USD, that represents a decline of $40 in the actual closing price. However, the adjusted close for December 5^{th} would change to $50 USD, making the gain $10 at the end of December 6th. The *volume* attribute is the number of shares traded on the given day.

Price Prediction We use the open, low, high, close, adjusted close, and volume of all preceding days as input to predict the close for a day, as explained in more details in section 4. We use the predicted price as a baseline to estimate the cost of vulnerabilities upon their disclosure. Upon examining the vendors in our dataset, we found 60 of them available through Yahoo Finance. Out of the 60 vendors, only 41 of vendors had vulnerabilities in our selected dataset. Out of those 41 vendors, 5 vendors had missing data attributes (e.g., blackberry had several "null"-valued attributes).

Press As a baseline for comparison with our results based on the approach used in the literature, we sample vulnerabilities reported in the media. We search for "software vulnerabilities in 2017" in *Forbes*, and *ZDNet*, and capture four vulnerabilities for comparison.

3.2 Assessing Vulnerability's Impact

To assess the impact of vulnerabilities, we separate our dataset by vendor. To find the effect of a vulnerability for the date on which the vulnerability was published, we look for the stock value on that particular date. It is worth noting that the stock markets do not open on weekends and holidays, making stocks unavailable on those days. For all dates with disclosed vulnerabilities whereby the stock data is unavailable, we approximate the open, low, high, close, adjusted close, and volume attributes in a linear relation with the last operating day and

the next operating day. For example, suppose the value on the last operating day, d_0, is x, the market was closed on days d_1, d_2, and d_3, and the value on next operating day, d_4, is y. We first calculate the number of days between d_0 and d_4, denoted by d (here, 3). We then approximate the values on days d_i for $i \in \{1, 2, 3\}$ as $d_i = x + \frac{i \times (y-x)}{d}$.

Finding the effect of a vulnerability is done by comparing the predicted stock price assuming the vulnerabilities did not exist with the actual price which takes the existence of the vulnerability into account. Therefore, we first predict a stock price for the no-vulnerability case and calculate the impact of the vulnerability's Abnormal Return on day i (AR_i for $i \in \{1, 2, 3\}$), where $\mathrm{AR}_i = R_i - \bar{R}$, such that Ri is the actual stock price on day i, and $\bar{R}$ is the expected stock without vulnerability (predicted). We then calculate the % of Abnormal Return on day i (PAR_i), where $i \in 1, 2, 3$, as $\mathrm{PAR}_i = \frac{\mathrm{AR}_i \times 100}{R_i}$.

Finally, we calculate the Overall (%) Abnormal Return on day i (OAR_i), where $i \in \{1, 2, 3\}$. For vendor $\{V_1, \ldots, V_m\}$ with vulnerability $\{v_1, \ldots, v_n\}$, the PAR values for a vulnerability v_j are denoted by PAR_i^j for $i \in \{1, 2, 3\}$. We calculate $\mathrm{OAR}_i^k = \sum_{j=1}^{n} \mathrm{PAR}_i^j$ on day i for a vendor V_k.

4 Prediction

The data of all vendors consists of the aforementioned features: date, open, close, high, low, volume and fractional change in the price from previous time step. All of these features, except date, are considered to predict the close value in the future. In order to increase the performance of the machine learning algorithm, data preprocessing is required. The general method for feature standardization is to consider the mean and standard deviation of each feature. In other words, feature standardization projects the raw data into a new space where each feature in the data has a mean and a standard deviation of zero and unit, respectively. This is, the mapping transforms the feature vector x into $z = \frac{x - \bar{x}}{\sigma}$, where $\bar{x}$ and σ, are the mean and standard deviation of the original feature vector x, respectively. These features are then fed into the nonlinear autoregressive neural network with exogenous factors (NARX) to predict the stock value of vendors.

4.1 NARX Neural Network

The NARX neural network, generally applied for prediction of the behavior of discrete-time nonlinear dynamical systems, is one of the most efficient tools of forecasting [31]. Unique characteristics of NARX provide accurate forecasts of the stock values by exploiting an architecture of recurrent neural network with limited feedback from the output neuron. In comparison with other architectures, which consider feedback from both hidden and output neurons, NARX is more efficient and yields better results [32]. Based on the NARX neural network model, the next value of the output at time t, $y(t)$, can be regressed on previous values of the output and exogenous input, represented using the following model:

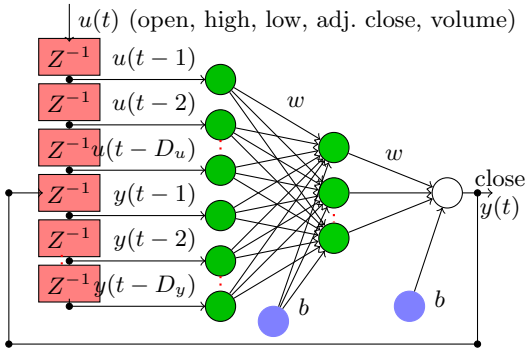

Fig. 2: General Structure of the NARX Neural Network

Table 1: NARX parameter settings.

Parameter	Value
Number of input neurons	Five
Number of output neurons	One
Transfer functions	tansig (hidden layer) purelin (output layer)
Training, validation, testing	70%, 15%, and 15%
Evaluation function	Mean squared error
Learning Algorithm	Levenberg-Marquardt

$$y(t) = f[u(t-1), ..., u(t-d_u); y(t-1), ..., y(t-D_y)],$$

where $u(t)$ and $y(t)$ are the input and output of the network at time t. d_u and d_y, are the lags of exogenous inputs and output of the system, and the function f is multi-layer feed forward network. The general architecture of the NARX neural network is shown in Fig. 2.

For each vendor, we divide the dataset into training, validation and test subsets (with 70%, 15%, and 15%, respectively). We use the training data to train a predictive model. The Mean Squared Error (MSE) is used to evaluate the performance of the corresponding models. The MSE is defined as:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_{t_i} - y_{p_i})^2,$$

where n is the number of samples. y_t and y_p are representing the actual value of the stock price and corresponding predicted value, respectively. A feed forward neural network with one hidden layer has been used as predictor function of the NARX. Levenberg-Marquardt (LM) back-propagation learning algorithm [33]

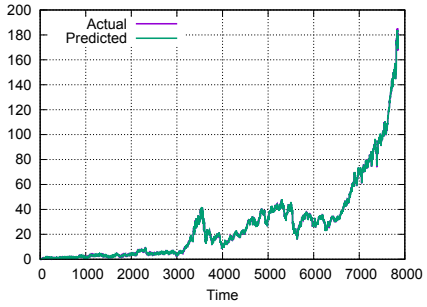

Fig. 3: Actual vs. Predicted: NARX.

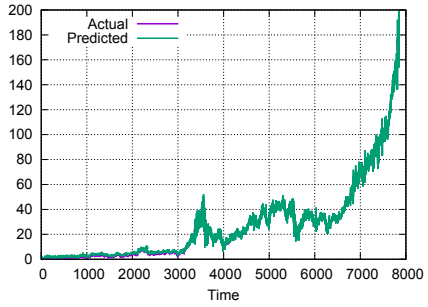

Fig. 4: Actual vs. Predicted: ARIMA.

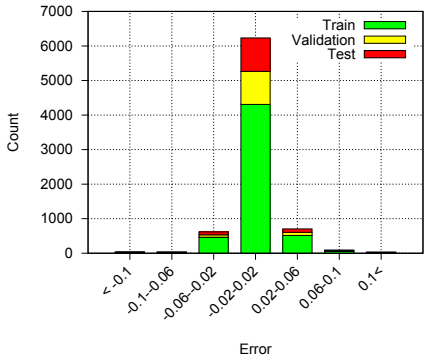

Fig. 5: Error Histogram of Adobe Stock.

has been employed to train the weights of the neural network. The specifications of the proposed NARX neural network are presented in Table 1.

Baseline for Comparison. In addition to the NARX neural network model, we also predicted the stock price of vendors using the Autoregressive Integrated Moving Average (ARIMA) model [34], one of the most popular time series prediction models, for comparison. To establish such a comparison with prior work using linear regression, we conducted the prediction for the stock price of one vendor, namely, Adobe. The AR portion of ARIMA signifies the variable to be predicted is regressed on its past values. Also, the MA portion in the ARIMA model indicates that the error in the regression model is a linear combination of error values in the past. The ARIMA model with external regressors, x, and for one-step ahead prediction can be represented by

$$y_p(t) - \phi_1 y_t(t-1) = \mu - \theta_1 e(t-1) + \beta(x(t) - \phi_1 x(t-1)),$$

where y_p and y_t are the predicted and actual prices of the stock, respectively. μ, θ, and ϕ are a constant, the MA coefficient, and the AR coefficient values.

The results are shown only for Adobe and for the rest of the vendors only the MSE is shown in Table 2. Fig. 3 depicts the actual and predicted stock

Table 2: Results for each Vendor. Vul. stands for vulnerability count and OAR_1, OAR_2, and OAR_3 stand for the average effect at day 1, 2, and 3 (percent), respectively. [(2)] Vendor names are abbreviated as follows: PAN=Palo Alto Networks, RWA=Rockwell Automation, TM=Trend Micro. ▲indicates that the vulnerabilities had no overall impact on vendor's stock value while ▼indicates that the stock of the vendor were impacted, overall.

Vendor	MSE	Vul.	OAR_1[(1)]	OAR_2[(1)]	OAR_3[(1)]	Vendor	MSE	Vul.	OAR_1[(1)]	OAR_2[(1)]	OAR_3[(1)]
Adobe	5.9E-4	494	▼0.65	▼0.37	▼0.50	Oracle	1.0E-3	130	▼0.48	▼0.81	▼1.51
Advantech	9.5E-4	9	▲0.61	▲0.89	▲0.96	Osram	7.8E-3	1	▲1.17	▼6.42	▼7.95
Apache	9.9E-4	37	▲0.60	▲0.98	▲1.17	PAN[(2)]	4.3E-3	2	▼1.09	▼1.13	▼8.54
Apple	2.8E-4	154	▲0.41	▲0.75	▲1.03	Redhat	1.6E-3	13	▲0.74	▲0.59	▲0.61
Atlassian	9.7E-3	4	▼3.85	▼3.86	▼3.12	RWA[(2)]	8.9E-4	5	▲1.47	▼0.87	▲0.06
Cisco	2.3E-3	111	▲0.10	▲0.33	▲0.42	Samsung	7.6E-3	10	▼0.08	▼0.08	▲2.95
Citrix	2.4E-3	9	▲0.14	▲0.01	▲0.57	Sap	2.3E-3	17	▲0.82	▲0.69	▲1.28
Facebook	1.1E-3	6	▲0.13	▼0.33	▲0.45	Schneider	3.1E-3	7	▼1.56	▼1.87	▼1.79
Fortinet	4.5E-3	7	▲0.37	▲0.19	▲0.92	Siemens	3.7E-3	14	▲0.51	▲0.83	▲0.32
GE	5.8E-4	3	▲0.12	▼0.58	▼0.39	Sophos	3.8E-3	3	▲1.72	▲1.87	▲0.89
Google	7.6E-4	410	▼0.08	▼0.21	▼0.08	Splunk	1.2E-2	1	▲0.88	▲3.17	▲1.11
Honeywell	4.3E-4	1	▼0.09	▲0.87	▲2.35	Symantec	1.3E-3	13	▲0.24	▲0.52	▲0.77
HP	7.6E-3	36	▲0.21	▲0.37	▲0.64	Teradata	3.6E-3	3	▼2.18	▼2.86	▼2.75
IBM	4.4E-4	51	▲0.22	▲0.32	▲0.26	TM[(2)]	9.3E-3	16	▼0.56	▼0.74	▲0.98
Juniper	6.3E-3	13	▼0.19	▼0.80	▼1.10	Vmware	6.1E-3	11	▲0.45	▲0.32	▲0.74
Lenovo	7.4E-3	9	▼0.75	▼1.12	▼0.55	Zyxel	5.2E-3	2	▲0.18	▼1.18	▲0.18
Microsoft	8.6E-4	279	▲0.45	▲0.39	▲0.56	Equifax	4.9E-4	1	▲1.52	▼14.02	▼24.19
Netapp	6.5E-3	4	▲1.08	▲0.76	▼1.19	Dow Jones	3.5E-4	1	▼0.08	▼0.34	▼0.03
Netgear	4.3E-3	14	▲1.18	▲1.61	▲0.10	Alteryx	4.8E-2	1	▼0.61	▼2.18	▼7.70
Nvidia	1.0E-3	38	▲0.56	▲1.46	▲4.39	Viacom	2.3E-3	1	▼1.60	▲0.60	▼0.62

price. The low value of the error strongly suggests that the NARX model can forecast the stock values with high accuracy. In addition, The error histogram is provided in Fig. 5, and shows that the majority of the instances are forecasted precisely. In Fig. 4, although visual representation suggests a weakness of fit with ARIMA in prediction the stock values, the difference in the value of MSE for these to models, 6.42 for ARIMA and 0.59 for NARX, quantitatively justifies the goodness of the proposed method over methods used in the literature.

5 Results

We experimented with a large number of vulnerabilities, meaning that multiple vulnerabilities could correspond to a single date. Therefore, the effect we see could be due to one or more vulnerabilities. For every vulnerability disclosure date and vendor, we calculate % Abnormal Return on days 0, 1, and 2 (AR_1, AR_2, and AR_3 respectively as described above). The results are presented in Table 2. The table contains the normalized MSE, count of the vulnerabilities, and Abnormal Return on days 1, 2, and 3 for every vendor (as described above). We observe that vulnerabilities had an adverse impact on the stock price of 17 out of the 36 vendors.

Table 4 represents a breakdown of vendors by industry and their likelihood of their stock being impacted by vulnerabilities. For the classification of indus-

tries, the software industry contains vendors such as Adobe, Apache, Atlassian, Google, VMware, Sap, Oracle, Redhat, and Alteryx. The device industry includes Advantech and Apple. The networking industry includes Cisco, Citrix, Netgear, and Zyxel. The security industry includes Fortinet, Juniper, Paloalto Networks, Symantec, and Trendmicro. The consumer product industry includes Rockwell Automation, Osram, Splunk, Schneider, Teradata, Facebook, Netapp, and Viacom. The electronics & hardware industry includes Lenovo, and Nvidia. Finally, the finance industry includes Equifax and Dow Jones. To assign a likelihood of an industry's stock price being impacted by vulnerabilities, we use Highly-Likely when the number of vendors with stock price affected negatively by the vulnerabilities in the given industry is larger than those not affected, Less-Likely otherwise; we use Equally-Likely when the number of vendors affected equals the number of vendors not affected.

We look at vulnerabilities from 10 vendors to find the reason for the nearly no-effect of vulnerabilities in some industries. We see that in every dataset there are a few dates which have no significant positive effect (from vendors perspective) on the market leading the results to be negative. By referring to the description of the vulnerabilities, we observe that:

1. Vulnerabilities affecting vendors' stock negatively are of critical severity (vulnerabilities with CVSS version 3 label of CRITICAL) while the rest were less severe (vulnerabilities with CVSS labels of HIGH or MEDIUM).
2. Vulnerabilities affecting vendors' stock price negatively have a combination of version 3 label of HIGH or CRITICAL, and a description containing phrases such as "denial of service", "allows remote attacker to read/execute", "allows context-dependent attackers to conduct XML External Entity XXE attacks via a crafted PDF", and "allows context-dependent attackers to have unspecified impact via an invalid character". Additionally, vulnerabilities description such as "allows authenticated remote attacker to read/execute", "remote attackers to cause a denial of service", and "allows remote attackers to write to files of arbitrary types via unspecified vectors" have little (on days 0, 1, and 2) to no effect on the stock price. Therefore, we can conclude that vulnerabilities involving unauthorized accesses have a higher cost, seen in their detrimental effect on the stock price.
3. Vulnerabilities with phrases such as "local users with access to' and "denial of service" in the description have no impact on the stock. Therefore, DoS attacks lacking confidentiality factor lead to no impact on stock value.

For the vulnerabilities gathered from the press, we followed the same steps. We found that these vulnerabilities have an adverse effect on vendor stock price in almost every case.

6 Statistical Significance

To understand the statistical significance of our results, we use the confidence interval of the observations as a guideline. Particularly, we measure the statistical

confidence of overall effect of vulnerabilities corresponding to a vendor on days 1, 2, and 3, respectively. Table 3 shows the confidence intervals (lower and upper limit) on days 1, 2, and 3, measured with 95% confidence.

95% Confidence Interval. 95% Confidence Interval (CI) is a range that contains the true mean of a population with 95% certainty. For a smaller population, the CI is almost similar to the range of the data, while only a tiny sample of data lies within the confidence interval for a large population. In our study, we have noticed that our data populations are diverse, where some vendors have a small number of samples, and others have larger number of samples. For example, Figure 6 – Figure 8 show the distribution of observations of effect for multiple example vendors and several vulnerabilities associated with each vendor. The shown histogram captures counts of the effect of vulnerabilities; the x-axis includes brackets of the effect (measured by OAR) and the y-axsis captures the count for the given effect. The diversity of the effect is well-captured by the count distribution; high severity impact is seen in a vendor where the counts are focused in the negative side of the interval, whereas lower (or no) impact is seen where the count focus is in the positive side. The confidence interval with 95% confidence for a given population (distribution) can be calculated as,

$$\text{CI} = \left(\bar{x} - 1.96\frac{\sigma}{\sqrt{n}}, \bar{x} + 1.96\frac{\sigma}{\sqrt{n}} \right),$$

where $\bar{x}$ is the mean of the population, σ is the standard deviation, and n is the number of samples in the population.

Putting it into perspective, while OAR_i, where $i \in \{1, 2, 3\}$, captures the overall effect of vulnerabilities corresponding to a vendor, the Confidence Interval (CI_i, where $i \in \{1, 2, 3\}$) gives the confidence for the effect to lie within its upper and lower bound. In Table 3, and by considering the data associated with Adobe, for example, we can say with 95% confidence that the confidence interval for the population, CI_i, contains the true mean, OAR_i. We also observe that:

1. Our OAR_i in Table 2 are within their respective confidence intervals, which means that our results reported earlier are statistically significant.
2. The true mean values for Adobe, Palo Alto Networks, Schneider Electric, and Teradata, on the day a vulnerability is disclosed, are bounded in negative intervals. Thus, the probability for a vulnerability having an effect on the day a vulnerability is disclosed on the vendor's stock price is highly likely.
3. The true mean for Oracle, Palo Alto Networks, Schneider Electric, and Zyxel on days after the day a vulnerability is disclosed are bounded in negative intervals. Thus, the probability for a vulnerability having a negative impact on days succeeding the day a vulnerability is disclosed on the vendor's stock price is highly likely.
4. The true mean for every vendor on the three days is bounded from below by negative value. Although the confidence intervals do not say anything about the percentage of population that would fall in the negative side of the interval, the lower bound indicate a likelihood that the population would have samples with negative effect on the vendor's stock. Thus, given the

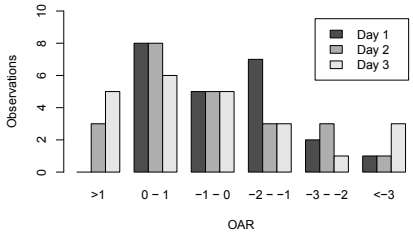

Fig. 6: Histogram of the effect of vulnerabilities on stock value: Adobe

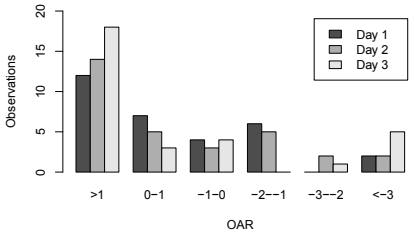

Fig. 7: Histogram of the effect of vulnerabilities on stock value: Apache

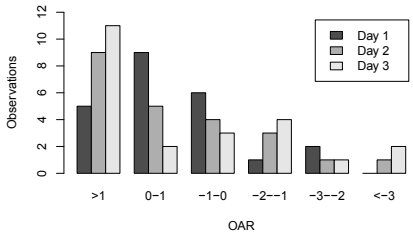

Fig. 8: Histogram of the effect of vulnerabilities on stock value: Apple

various vulnerabilities on a specific vendor, it is likely that some of those vulnerabilities would have a negative effect on the vendor's stock value, even though the overall effect (measured by the mean) would be nullified. This, as well, is well captured in our analysis.

7 Discussion and Comparison

There has been several works dedicated to understanding the hidden cost of software vulnerabilities in the literature, which we discuss in the following across multiple aspects by comparison.

7.1 Comparison of Findings with Prior Work

The prior work has made various conclusions concerning the effect of the software vulnerabilities, and whether they are associated with a certain feature of those vulnerabilities, including correlation with types, publicity, etc. In the following, we compare our work and findings with the prior work across multiple factors, including vulnerability type, publicity, data source, methodology, and sector.

Table 3: Statistical confidence for each Vendor. OAR_1, OAR_2, and OAR_3 stand for the average effect at day 1, 2, and 3 (percent), respectively. CI_i is the confidence interval for day_i, where i $\epsilon\{1, 2, 3\}$. $^{(2)}$ Vendor names are abbreviated; PAN=Palo Alto Networks, RWA=Rockwell Automation, TM=Trend Micro.

Vendor	CI_1		CI_2		CI_3		Vendor	CI_1		CI_2		CI_3	
	Low	High	Low	High	Low	High		Low	High	Low	High	Low	High
Adobe	-1.10	-0.20	-0.96	0.22	-1.23	0.23	Oracle	-1.08	0.12	-1.19	-0.43	-2.10	-0.92
Advantec	-0.96	2.18	-2.20	3.98	-3.02	4.94	PAN$^{(2)}$	-1.80	-0.37	-2.10	-0.15	-24.23	7.15
Apache	-0.17	1.45	-0.40	2.36	-0.64	2.98	Redhat	-0.19	1.68	-0.33	1.51	-0.64	1.86
Apple	-0.25	1.07	-0.11	1.62	-0.17	2.24	RWA$^{(2)}$	-0.19	3.13	-2.18	2.00	-1.67	1.79
Atlassian	-2.05	0.53	-3.41	1.62	-2.77	2.50	Samsung	-0.21	0.06	-0.21	0.06	-3.07	8.96
Cisco	-0.22	0.41	-0.20	0.85	-0.17	1.02	Sap	-0.31	1.94	-0.57	1.94	-0.10	2.66
Citrix	-0.46	0.75	-0.93	0.94	-0.69	1.83	Schneider	-2.95	-0.17	-3.36	-0.37	-4.17	0.58
Facebook	-0.38	0.63	-0.74	0.08	-2.37	3.27	Siemens	-0.19	1.22	-0.60	2.26	-1.10	1.73
Fortinet	-1.04	2.98	-0.76	2.66	-1.48	3.07	Sophos	-0.19	3.64	0.77	2.96	-1.03	2.80
GE	-1.05	1.30	-1.54	0.37	-2.28	1.50	Symantec	-0.20	0.69	-0.05	1.09	-0.09	1.63
Google	-0.41	0.25	-0.76	0.34	-0.75	0.60	Teradata	-2.50	-1.86	-4.63	-1.10	-8.29	2.79
HP	-0.38	0.79	-0.35	1.09	-0.34	1.63	TM$^{(2)}$	-1.71	0.60	-1.90	0.42	-0.41	2.37
IBM	-0.04	0.48	-0.11	0.74	-0.17	0.69	Vmware	-0.51	1.41	-0.79	1.42	-0.86	2.34
Juniper	-1.66	1.29	-2.38	0.79	-3.57	1.37	Zyxel	-0.52	0.88	-1.42	-0.95	-2.27	2.64
Lenovo	-1.55	0.05	-2.67	0.42	-2.69	1.59	Nvidia	-0.49	1.60	-0.57	3.49	1.10	7.67
Microsoft	-0.03	0.92	-0.31	1.08	-0.20	1.33	Netgear	-0.16	2.52	0.21	3.00	-2.28	2.48
Netapp	-0.44	2.59	-0.27	1.80	-4.13	1.74							

Confidentiality vs. non-confidentiality vulnerabilities (confirmation). Campbell *et al.* [15] observed a negative market reaction for information security breaches involving unauthorized access to confidential data, and reported no significant reaction to non-confidentiality related breaches. Through our analysis, we had a similar conclusion. Particularly, we found that vulnerabilities affecting vendor's stock negatively have descriptions containing phrases indicating confidentiality breaches, such as "denial of service", "allows remote attacker to read/execute", "allows context-dependent attackers to conduct XML External Entity XXE attacks via a crafted PDF", and "allows context-dependent attackers to have unspecified impact via an invalid character".

How publicity affects price (contradiction). There has been several works in the literature on attempting to understand how the coverage by media and other forms of publicity for viruses and data breaches affect the stock value of a given vendor associated with such vulnerabilities. For example, Hovav and D'Arcy [10] demonstrated that virus-related announcements do not impact stock price of vendors. Our results partly contradict their claims, as we show that vulnerabilities impact the stock value a vendor, sometimes significantly (negatively), regardless to whether such vulnerabilities are announced or not.

Table 4: Per industry stock impact likelihood analysis.

Industry	Likeliness
Software	Highly Likely
Consumer Products	Highly Likely
Finance	Highly Likely
Security	Equally Likely
Electronics & Hardware	Equally Likely
Conglomerate	Less Likely
Device	Less Likely
Networking	Less Likely

Data source and effect (broadening scopes). Goel *et al.* [14] and Telang and Wattal [13] estimated the impact of vulnerabilities on the stock value of a given vendor by calculating a Cumulative Abnormal Rate (CAR) and using a linear regression model. Their results are based on security incidents: while both gather data from the press, Telang and Wattal [13] also use a few incidents from Computer Emergency Response Team (CERT) reports. On the other hand, we consider a wide range of vulnerabilities regardless of being reported by the press. Our results show various trends and indicate the dynamic and wide spectrum of effect of vulnerabilities on the stock price of vendors.

Methodology (Addressing caveats of prior work). The prior work shows the impact of vulnerabilities using CAR, which aggregates AR's on different days. However, we refrain from using CAR because of the following. First, CAR does not effectively capture the impact of a vulnerability, due to information loss by aggregation. For example, CAR would indicate no-effect if the magnitude (upward) of one or more days analyzed negate the magnitude (downward) of other days. Second, we consider a vulnerability as having had an impact if the stock shows a downward trend on d_1, d_2, or d_3, irrespective of the magnitude. Third, our results, through a rigorous analysis are statistically significant. To demonstrate the caveats of CAR and show the benefits of our approach in capturing a better state of the effect of vulnerabilities on the stock price, we consider both Samsung and Equifax in Table 2. On the one hand, the impact of vulnerability on Equifax on days 2 and 3 was significant (-14.02 and -24.09 vs. +1.52 on day 1), where CAR would capture the effect. On the other hand, such an effect would not be captured by CAR with Samsung (-0.08 and -0.08 on days 1 and 2 vs. +2.95 on day 3). Our approach, however, considers the effect of the vulnerability the stock price over the different days separately (and does not lose information due to aggregation).

Sector-based analysis. A general hypothesis is that the cost of security and vulnerabilities on vendors is sector-dependent. One of the main shortcomings of the prior work, however, is that it overlooks analyzing the cost based on sectors of the software industry. By classifying vendors based a clear industry sector, our results show the likelihood of effect to be high in software and consumer product industry, while the likelihood is less in the device, networking or conglomerate

industries. Table 4 further highlights the industries with highest losses, by tracking losses by individual vendors. Although Table 2 shows that a vulnerability may or may not have an effect on its vendor's stock price, Table 3 shows that individual vulnerabilities may affect the stocks' value.

Shortcomings. In this study we found a significant effect of vulnerabilities on a given day and limited ourselves to the second day after the release of the vulnerability in order to minimize the impact of other factors. However, other factors may affect the stock value than the vulnerability, making the results unreliable, and highlight the correlational-nature of our study (as opposed to causational). Eliminating the effect of those factors, once known, is an open question. Furthermore, apart from the effect on stock, a vendor may sustain other hidden and long-term losses, such as consumers churn (switching to other products or vendors), loss of reputation, and internal losses (such as man-hour for developing remedies), which we do not consider in our evaluation, and open various directions for future work.

7.2 Breaches and Disclosure

Our analysis of the vulnerabilities show that while vulnerabilities may or may not have an impact on the stock price, a vulnerability reported by the press is highly likely to impact the stock price. The diverse results for the vulnerabilities collected from NVD are explained by the diverse severity of the vulnerabilities, whereas 1) the press may report on highly critical vulnerabilities that are more likely to result in loss, or 2) the reported vulnerabilities in the press may create a negative perception of the vendor leading to loss in their stock value. This, as a result, led many vendors to not disclose vulnerabilities in order to cope with bad publicity. For example, Microsoft did not disclose an attack on its bug tracking system in 2013 [35], demonstrating the such a behavior in vendors when dealing with vulnerabilities [36]. Recent reports also indicate a similar behavior by Yahoo when their online accounts were compromised, or by Uber when their employees and users personal information were leaked. More broadly, a recent survey of 343 security professionals worldwide indicated that the management of 20% of the respondents considered cyber-security issues a low priority, alluding to the possibility of not disclosing vulnerabilities even when they affect their systems [37].

8 Conclusion and Future Work

We perform an empirical analysis on vulnerabilities from NVD and look at their effect on vendor's stock price. Our results show that the effect is industry-specific, and depends on the severity of the reported vulnerabilities. We also compare the results with the vulnerabilities found in popular press: while both vulnerabilities affect the vendor's stock, vulnerabilities reported in the media have a much more adverse effect. En route, we also design a model to predict the stock price with high accuracy. Our work is limited in a sense that we do not consider other

external factors affecting the stock or internal factors affecting long term users behavior and deriving vulnerabilities cost. Exploring those factors along with regional differences in effect will be our future work.

9 Acknowledgement

This work is supported in part by NSF grant CNS-1809000 and NRF grant NRF-2016K1A1A2912757. Part of this work has been presented as a poster at ACM AsiaCCS 2018 [38].

References

1. A. Wang, A. Mohaisen, W. Chang, and S. Chen, "Delving into internet ddos attacks by botnets: Characterization and analysis," in *Proceedings of the 45th International Conference on Dependable Systems and Networks (DSN)*, Rio de Janeiro, Brazil, 2015, pp. 379–390.
2. ——, "Measuring and analyzing trends in recent distributed denial of service attacks," in *Proceedings of the 17th International Workshop on Information Security Applications (WISA)*, 2016, pp. 15–28.
3. J. Spaulding, D. Nyang, and A. Mohaisen, "Understanding the effectiveness of typosquatting techniques," in *Proceedings of the 5th ACM/IEEE Workshop on Hot Topics in Web Systems and Technologies*, 2017, p. 9.
4. G. Tassey, "The economic impacts of inadequate infrastructure for software testing," *National Institute of Standards and Technology, RTI Project*, vol. 7007, no. 011, 2002.
5. J. Strasburg and J. Bunge, "Loss swamps trading firm, knight capital searches for partner as tab for computer glitch hits $440 million," *Wall Street Journal (Online). Retrieved from http://search. proquest. com/docview/1033163975*, 2012.
6. J. Berr, ""wannacry" ransomware attack losses could reach $4 billion," May 2017. [Online]. Available: http://cbsn.ws/2yYjif2
7. "The cost impact of major virus attacks since 1995." [Online]. Available: http://www.computereconomics.com/article.cfm?id=936
8. L. Geppert, "Lost radio contact leaves pilots on their own," *IEEE spectrum*, vol. 41, no. 11, pp. 16–17, 2004.
9. G. Jarrell and S. Peltzman, "The impact of product recalls on the wealth of sellers," *Journal of Political Economy*, vol. 93, no. 3, pp. 512–536, 1985.
10. A. Hovav and J. D'arcy, "Capital market reaction to defective it products: The case of computer viruses," *Computers & Security*, vol. 24, no. 5, pp. 409–424, 2005.
11. S. Romanosky, D. Hoffman, and A. Acquisti, "Empirical analysis of data breach litigation," *Journal of Empirical Legal Studies*, vol. 11, no. 1, pp. 74–104, 2014.
12. G. Spanos and L. Angelis, "The impact of information security events to the stock market: A systematic literature review," *Computers & Security*, vol. 58, pp. 216–229, 2016.
13. R. Telang and S. Wattal, "An empirical analysis of the impact of software vulnerability announcements on firm stock price," *IEEE Transactions on Software Engineering*, vol. 33, no. 8, pp. 544–557, 2007.
14. S. Goel and H. A. Shawky, "Estimating the market impact of security breach announcements on firm values," *Information & Management*, vol. 46, no. 7, pp. 404–410, 2009.

15. K. Campbell, L. A. Gordon, M. P. Loeb, and L. Zhou, "The economic cost of publicly announced information security breaches: empirical evidence from the stock market," *Journal of Computer Security*, vol. 11, no. 3, pp. 431–448, 2003.
16. H. Cavusoglu, B. Mishra, and S. Raghunathan, "The effect of internet security breach announcements on market value: Capital market reactions for breached firms and internet security developers," *International Journal of Electronic Commerce*, vol. 9, no. 1, pp. 70–104, 2004.
17. I. Bose and A. C. M. Leung, "Do phishing alerts impact global corporations? a firm value analysis," *Decision Support Systems*, vol. 64, pp. 67–78, 2014.
18. F. Li and V. Paxson, "A large-scale empirical study of security patches," in *Proceedings of the 24th ACM Conference on Computer and Communications Security (CCS)*, Dallas, TX, Oct.–Nov. 2017, pp. 2201–2215.
19. V. H. Nguyen and F. Massacci, "The (un) reliability of nvd vulnerable versions data: An empirical experiment on google chrome vulnerabilities," in *Proceedings of the 8th ACM Symposium on Information, Computer and Communications Security (ASIACCS)*, Sydney, Australia, Mar. 2013, pp. 493–498.
20. S. Christey and B. Martin, "Buying into the bias: Why vulnerability statistics suck," *BlackHat, Las Vegas, USA, Technical Report*, vol. 1, 2013.
21. S. Romanosky, R. Telang, and A. Acquisti, "Do data breach disclosure laws reduce identity theft?" *Journal of Policy Analysis and Management*, vol. 30, no. 2, pp. 256–286, 2011.
22. L. A. Gordon, M. P. Loeb, and L. Zhou, "The impact of information security breaches: Has there been a downward shift in costs?" *Journal of Computer Security*, vol. 19, no. 1, pp. 33–56, 2011.
23. A. Kar, "Stock prediction using artificial neural networks," *Department of Computer Science and Engineering, IIT Kanpur*, 1990.
24. S. Farhang, A. Laszka, and J. Grossklags, "An economic study of the effect of android platform fragmentation on security updates," *arXiv preprint arXiv:1712.08222*, 2017.
25. "National vulnerability database (nvd), url=https://nvd.nist.gov/."
26. "Symbol lookup from yahoo! finance." [Online]. Available: https://finance.yahoo.com/lookup/
27. "CVE - common vulnerabilities and exposures (cve)." [Online]. Available: https://cve.mitre.org/
28. "Common weakness enumeration." [Online]. Available: https://cwe.mitre.org/
29. "Common vulnerability scoring system sig." [Online]. Available: https://www.first.org/cvss/
30. CVSS version 3. [Online]. Available: https://www.first.org/cvss/cvss-v30-user_guide_v1.1.pdf
31. J. L. Elman, "Finding structure in time," *Cognitive science*, vol. 14, no. 2, pp. 179–211, 1990.
32. B. G. Horne and C. L. Giles, "An experimental comparison of recurrent neural networks," in *Proceedings of the Advances in Neural Information Processing Systems 7, [NIPS Conference]*, 1994, pp. 697–704.
33. J. J. Moré, "The levenberg-marquardt algorithm: implementation and theory," in *Numerical analysis*, 1978, pp. 105–116.
34. G. E. Box and D. A. Pierce, "Distribution of residual autocorrelations in autoregressive-integrated moving average time series models," *Journal of the American Statistical Association*, vol. 65, no. 332, pp. 1509–1526, 1970.
35. J. Menn, "Exclusive: Microsoft responded quietly after detecting secret database hack in 2013," Oct 2017. [Online]. Available: http://reut.rs/2ysNpw2

36. “A social science approach to information security.” [Online]. Available: http://bit.ly/2l7IefL
37. B. Violino, “Data breaches rising because of lack of cybersecurity acumen,” Dec 2017. [Online]. Available: http://bit.ly/2CbIQKR
38. A. Anwar, A. Khormali, and A. Mohaisen, “Poster: Understanding the hidden cost of software vulnerabilities: Measurements and predictions,” in *Proceedings of the 13th ACM Symposium on Information, Computer and Communications Security (ASIACCS)*, Incheon, Korea, Jun. 2018.