International Journal of Electrical and Computer Engineering (IJECE)
Vol. 11, No. 5, October 2021, pp. 4381~4391
ISSN: 2088-8708, DOI: 10.11591/ijece.v11i5.pp4381-4391

Forecasting number of vulnerabilities using long short-term neural memory network

Mohammad Shamsul Hoque[1], Norziana Jamil[2], Nowshad Amin[3], Azril Azam Abdul Rahim[4], Razali B. Jidin[5]
[1,2,5]College of Computing and Informatics, Universiti Tenaga Nasional, Malaysia
[3]Institute of Sustainable Energy, Universiti Tenaga Nasional, Malaysia
[4]Tenaga Nasional Berhad, Kuala Lumpur, Malaysia

Article Info

Article history:

Received Aug 9, 2020
Revised Mar 12, 2021
Accepted Mar 27, 2021

Keywords:

Information security
Long short-term memory network
Recurrent neural network
Supervised machine learning
Threat intelligence
Time series
Vulnerability prediction model

ABSTRACT

Cyber-attacks are launched through the exploitation of some existing vulnerabilities in the software, hardware, system and/or network. Machine learning algorithms can be used to forecast the number of post release vulnerabilities. Traditional neural networks work like a black box approach; hence it is unclear how reasoning is used in utilizing past data points in inferring the subsequent data points. However, the long short-term memory network (LSTM), a variant of the recurrent neural network, is able to address this limitation by introducing a lot of loops in its network to retain and utilize past data points for future calculations. Moving on from the previous finding, we further enhance the results to predict the number of vulnerabilities by developing a time series-based sequential model using a long short-term memory neural network. Specifically, this study developed a supervised machine learning based on the non-linear sequential time series forecasting model with a long short-term memory neural network to predict the number of vulnerabilities for three vendors having the highest number of vulnerabilities published in the national vulnerability database (NVD), namely microsoft, IBM and oracle. Our proposed model outperforms the existing models with a prediction result root mean squared error (RMSE) of as low as 0.072.

Corresponding Author:

Mohammad Shamsul Hoque
College of Computing and Informatics
Universiti Tenaga Nasional, Malaysia
Email: shams_uttara@gmail.com

1. INTRODUCTION

Vulnerability reflects the weakness of a system which leads to most security breaches. In some cases of successful cyber-attacks, the hackers would usually exploit zero-day vulnerabilities before their existence is noticed by the security administrator [1], [2]. The importance of security focus has dramatically increased, and therefore past vulnerability trends and reviews become more valuable to be considered in the acquisition of a new software system [3]. Machine learning [4] is a subfield of computer science and is an application of artificial intelligence (AI) that "gives computers the ability to learn without being explicitly programmed" [5]. Data modelling for predicting vulnerabilities which thereby creates deployable machine learning models is a tool that can be used as a proxy to predict the number of future vulnerabilities or to predict vulnerabilities most likely likelihood to be exploitable. The authors in [5] and [6] showed that addressing security issues during software development phase is an arduous task since project managers would usually focus on cost and the timely delivery of products, thus resulting in the high probability of the existence of vulnerabilities.

During software development phase is an arduous task since project managers would usually focus on cost and the timely delivery of products, thus resulting in the high probability of the existence of vulnerabilities.

Researchers have used various forms of vulnerability databases in developing models for vulnerabilities disclosure trends. The aim of most of these researchers is in finding the techniques in developing models that could predict the number of future product vulnerabilities using their historical data through regression and time series analysis [7]-[15]. However, these models suffer in various extents from the limitations stemming from their underlying assumptions, use of machine learning algorithms and high expectation for accuracy. The utilization of the linear algorithm is unable to deduce the underlying non-linear aspect of the data. The neural network model application for example, traditionally follows the black box approach that are not mathematically tractable and cannot be easily interpreted. Besides that, in non-linear models other than the long short-term memory network, the number of inputs has to be selected beforehand, thus making learning impossible for functions that depend on historical input that took place a long time ago. Since security activities for software and systems are highly resource savvy, the models are therefore expected to have high accuracy of vulnerability prediction by the vendors, end users as well as businesses [16]. In this research, the prediction of the number of future vulnerabilities is taken as a supervised sequential time series forecasting problem, and makes the following contributions:

- Creation of a new, larger dataset for each selected vendor (Microsoft, Oracle and IBM) using a novel technique of feature aggregation obtained from open-source datasets which contains a lot more examples to efficiently train the LSTM network. These datasets contain a lot more examples compared to the previous work [15]. Our dataset (created by aggregating the "Published Date feature") for Microsoft vendor for example, has 1030 examples compared to [15] where the dataset was prepared based on a monthly aggregation and has approximately 252 examples for each. To the best of our knowledge, these are the first datasets with such feature created by grouping based on the published date to be utilized for vulnerability prediction model.
- Utilization of deep learning algorithm called the long short-term memory (LSTM), a variant of the recurrent neural network (RNN), known for having the unique capability in retaining past information of series in calculating the activation functions and weights in machine learning modelling to forecast future values with higher accuracy. To the best of our knowledge, this research is the first to utilize the long short-term memory neural network with the national vulnerability dataset in developing a machine learning model to forecast the number of future vulnerabilities.
- The proposed model in this research has achieved the accuracy (root men squared error) value of as low as 0.072 which has outperformed the existing models in terms of prediction accuracy and following distribution trends.

The rest of the paper is organized is being as: Section 2 describes the related work while. Section 3 describes the dataset and method used in the analysis. Section 4 describes the advantages of the LSTM model in time series prediction, section 5 describes the results and analysis, and finally section 6 describes the conclusion and future works.

2. RELATED RESEARCH

Lyu and Lyu [9] surveyed software defect detection processes using software reliability growth models. Anderson proposed the Anderson Thermodynamic (AT) time-based vulnerability discovery which is considered as a pioneer in such research [16]. Alhazmi and Malaiya proposed a time-based application of software reliability growth modelling (SRGM) in predicting the number of vulnerabilities, and later have also proposed another logistic regression model for Windows 98 and NT 4.0 in predicting the number of undiscovered vulnerabilities [17]. Rescola proposed two time-based trend models, namely the linear model (RL) and the exponential model (RE) to estimate future vulnerabilities [18]. Kim [19] proposed a new Weibull distribution-based vulnerability discovery model (VDM) which was compared with [20], and found that their model performed better in many cases. There are also several other studies that worked further based on the existing VDMs for various software packages with the aim of improving the vulnerability discovery rate and the prediction of future vulnerability count [21]-[28].

Movahedi *et al.* [29] developed nine common vulnerability discovery models (VDMs) which were compared with a nonlinear neural network model (NNM) over a prediction period of three years. The common VDMs are the NHPP power-law gamma-based VDM, Weibull-based VDM, AML VDM, normal-based VDM, rescorla exponential (RE), rescorla quadratic (RQ), younis folded (YF) and linear model (LM). These models use the NVD dataset with the feedforward NNM with a single hidden layer in forecasting four well-known OSs and four well-known web browsers and assessed the models in terms of prediction accuracy and prediction bias. The results showed that in terms of prediction accuracy, the NNM has outperformed the

VDMs in all cases while regarding the overall magnitude of bias, the NMM provided the smallest value against seven common VDMs out of eight.

In recent times, Pokhrel [15] proposed a vulnerability prediction model based on the non-linear approach using the time series analysis. They utilized the NVD database for the auto regressive moving average (ARIMA), artificial neural network (ANN), and support vector machine (SVM) to develop the prediction models and selected the operating systems such windows 7, Mac OS X, and Linux Kernel for the experiments. The examples of the result show that the best model for windows 7 was produced by SVM with symmetric mean absolute error (SMAPE), mean absolute error (MAE), and root mean squared error (RMSE) values of 0.12, 3.15 and 3.58 respectively. However, since nonlinearity is a common trend in vulnerability disclosure, traditional time series-based modelling may always have limited capability in attaining high prediction accuracy. In our study, the number of vulnerability predictions is modelled with newly-created datasets using the sequential LSTM network, and the prediction accuracy was found to have improved in comparison to other datasets such as the recent one by [15].

3. METHOD

Vulnerability data were extracted using a custom-coded web scrapper to dump the data for the top-three vendors from the MITRE's CVE website [16] spanning between 1997 to 2019. The number of vulnerabilities for Microsoft, Oracle and IBM are 6814, 6115 and 4679 respectively as shown in Figures 1 and 2. A new dataset was then created for each vendor by grouping it according to the "Published Date" attribute. Another attribute called the "CVE ID" was aggregated as the size of each group, and the dataset was later chronologically organized (in an ascending order) in preparing for the time series. Therefore, these datasets contain a lot more examples to train the deep learning algorithms compared to the previous works. For example, our dataset for the Microsoft vendor contained 1090 examples whereas in [16] the dataset was prepared according to the monthly aggregation and contained approximately 252 examples for each dataset.

Top 50 Vendors By Total Number Of "Distinct" Vulnerabilities

Go to year: 1999 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019 All Time Leaders

	Vendor Name	Number of Products	Number of Vulnerabilities	#Vulnerabilities/#Products
1	Microsoft	529	6814	13
2	Oracle	644	6115	9
3	IBM	1064	4679	4

Figure 1. Top-three vendors by vulnerability count (snipped from [30])

#	CVE ID	CWE ID	# of Exploits	Vulnerability Type(s)	Publish Date	Update Date	Score	Gained Access Level	Access	Complexity	Authentication	Conf.	Integ.	Avail.
1	CVE-2019-11397	22		Dir. Trav. File Inclusion	2019-05-14	2019-05-16	4.0	None	Remote	Low	Single system	Partial	None	None
GetFile.aspx in Rapid4 RapidFlows Enterprise Application Builder 4.5M.23 (when used with .NET Framework 4.5) allows Local File Inclusion via the FileDesc parameter.														
2	CVE-2019-1378	269			2019-10-10	2019-10-15	7.2	None	Local	Low	Not required	Complete	Complete	Complete
An elevation of privilege vulnerability exists in Windows 10 Update Assistant in the way it handles permissions.A locally authenticated attacker could run arbitrary code with elevated system privileges, aka 'Windows 10 Update Assistant Elevation of Privilege Vulnerability'.														

Figure 2. Attributes of dataset (for example Microsoft) (snipped from [30])

The whole research method consists of three phases. The first phase deals with data vulnerability for the top-three vendors according to the number of vulnerabilities collected from the open-source national vulnerability database with a custom-built web scrapper in a csv format. The second phase involves data preparation, cleaning and engineering which covers the bulk of the activities common to usual data modelling project. Major activities include the analysis of distribution characteristics such as stationarity, seasonality and linearity, cleaning of data, feature engineering, as well as data wrangling and aggregation in creating the larger novel training datasets. The third phase involves the experimentation and result analysis where the existing best performing model [16] was set as the baseline for comparison and improvement purposes. The long short-term memory network was utilized for the modelling, and parameter optimization was performed through grid search. The following Figure 3 shows the overall view of the processes of this research:

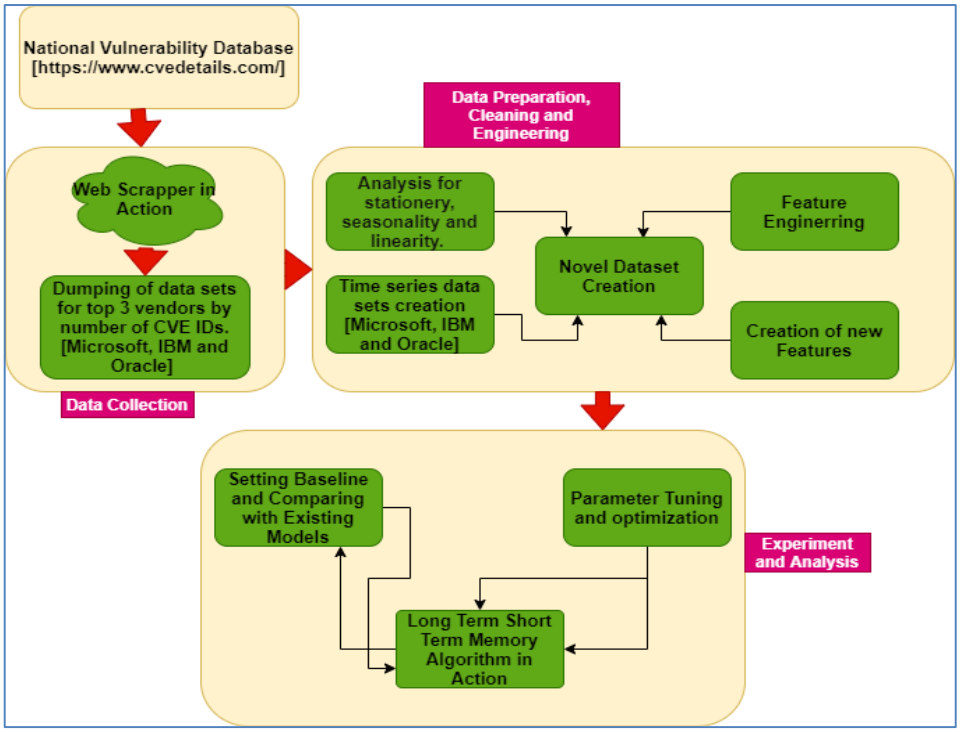

Figure 3. Method of the research

4. ADVANTAGES OF THE LONG SHORT-TERM MEMORY MODEL IN TIME SERIES FORECASTING

A special ANN called the recurrent neural network (RNN) is designed to utilize loops to persist information based on previously learnt knowledge. These looped networks work on each node in a sequence of series by performing the same processing techniques and are hence termed as recurrent. The RNNs maintain memory cells to capture information from past data points in the sequence as shown in Figure 4. In terms of modelling dependencies between two points in a sequence, the RNN models would usually perform better than NNMs. When applying the NNMs, in most cases, it is required to choose the length of the input (number of inputs) beforehand. Thus, the algorithm is unable to learn the information of a function that is dependent on data points of long time ago in a sequence. However, the RNN can avoid this drawback since it is able to store information from data points of long time ago.

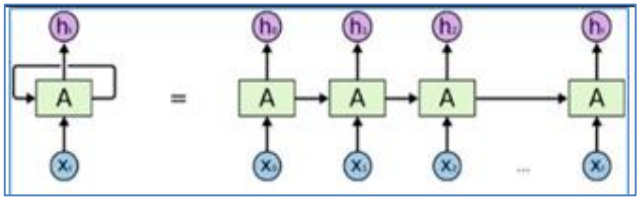

Figure 4. An unrolled recurrent neural network (Source: [31])

However, one significant drawback of the RNNs is that when the data sequence starts to increase, the network tends to lose information on historical context over time. A variant of the RNN, called the long short-term memory network (LSTM) can solve this problem since it maintains the cells for keeping information for the previous nodes irrespective of the sequence size. An LSTM network maintains [32] three or four gates where the input, output and forget gates are common as shown in Figure 5.

The notations *t, C and h* indicate one step in time, cell state and the hidden state respectively. The gates i (input), f (forget) and o (output) that are most usually modelled with a sigmoid layer (values ranging between 0-1) help in protecting and controlling the addition and removal of information in a cell state. The

LSTMs therefore try to overcome the shortcomings of the RNN models regarding the handling of long-term dependencies by mitigating the issue when the weight matrix of the neurons becomes too small (which may lead to the vanishing of the gradients) or too large (which may cause the gradients to explode). The following steps shows the mathematical functions [33], [34] of a long short-term memory neural network for typical usage, such as for this research experiment:

– Stage 1

For a forward pass in an LSTM block, if xt is considered as an input vector at a time t, N and M are the LSTM block number and input number respectively, an LSTM layer can have four weights:

Weights for input: Wz, Wi, Wf, Wo ∈ R N×M
Weights for peephole: pi, pf, po ∈ R N
Weights for recurrent: Rz, Ri, Rf, Ro ∈ R N×N and
Weights for bias: bz, bi, bf, bo ∈ R N
With reference to the weights above, the formula for vector in a forward pass for an LSTM layer are as follows, (e σ, g and h are activation functions):
z- t = Wz xt + Rz y t−1 + bz
zt = g (z-t) [block for input]
i-t = Wi xt + Ri y t−1 + pi • c t−1 + bi
it = σ (¯i t) [gate for input]
t-t = Wf xt+Rf yt-1 + pf • ct-1 + bf
ft = σ (f-t) [gate for forget]
ct = zt •it + ct-1 • ft [LSTM cell]
o-t = Wo xt + Ro yt-1+po • ct + bo
ot = σ (o-t) [gate for output]
yt = h (ct) • ot [output for block]

Typically, logistic sigmoid is the activation function at the LSTM gate while the hyperbolic tangent is the activation function at the block input and output.

– Stage 2

After the forward pass functions are computed, the following delta functions are computed inside the LSTM block for backpropagation through time stamps:

Δy-t = Δt + Rz T δzt+1 + Rt T δit+1+ RfT δft+1+ RoT δot+1
Δo-t = δyt • h(ct) • σ/(o-t)
Δct = δyt • ot • h/(ct) + po • δo-t + pi • δi-t+1 + pf • δf-t+1 + δct+1 • ft+1
Δf-t = δct • ct-1 • σ/(f-t)
Δit = δct • zt • σ/(i-t)
Δz-t = δct • it • g/(z-t)

Here Δt denotes the vector of deltas propagated from the upper layers to the downward layers. The value E corresponds to ∂E /∂yt when treated as a loss function. Input deltas are only required when the lower layer of the input layer requires training, and in such situation the value of the delta for the input is:

Δxt = WzT δz-t + WiT δi-t + WfT δf-t + WoT δo-t

– Stage 3

At the final stage, the gradients to adjust the weights are computed with the following functions, whereas ¤ denotes any of the vectors z, I, f and o, and (¤ 1. ¤ 2) are the outer products of the corresponding two vectors:

δW* = ∑t=0T (δ¤ t, xt) δpi = ∑t=0T-1ct • δi-t+1
δR* = ∑t=0T (δ¤ t+1, yt) δpf = ∑t=0T-1ct • δf-t+1
δb* = ∑t=0T δ¤ t δpo = ∑t=0T-1ct • δo-t+1

The RNNs, unlike the ARIMA, have the capability to learn nonlinearities in a data sequence, and specialized nodes such as the LSTM nodes could handle that even more efficiently. One requirement of the LSTM network is that a long sequence is needed in learning past dependencies and for this reason this research therefore aims to create a long sequence as described in the previous section.

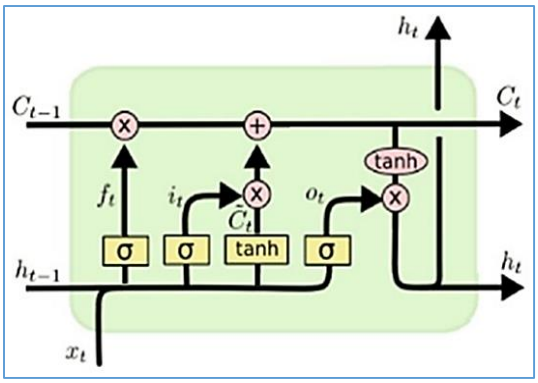

Figure 5. An LSTM node (Source: [31])

5. RESULTS AND ANALYSIS

Prior to applying a time series dataset into the machine learning algorithm, it is important to analyse the stationarity. The initial null hypothesis is that the time series at hand is non-stationary. Figures 6-8 show the rolling stats plots (red lines) where it can be observed that the number of vulnerabilities is low and stable at the beginning of time but shows spikes and lows as time grows. Generally, there will be an upward trend in the number of vulnerabilities over the course of time. However, a reliable trend or seasonality which does not demonstrate hikes and falls with a clear trend or seasonality signifies that it is a stationary sequence. The augmented dickey fuller test (ADF) with log transformation shows that the test statistical values as shown in Figures 9-11 are greater than any of the critical values, thus leading to the rejection of the null hypothesis (the data distribution is not stationary) in all cases. Although unlike traditional algorithms, the ARIMA 'stationarity' is less brittle with LSTM (RNN), the stationarity of the series will be sufficient for its performance. Furthermore, if the series is not stationary, it is then much safer for a transformation to be performed in making the data stationary and then train the LSTM.

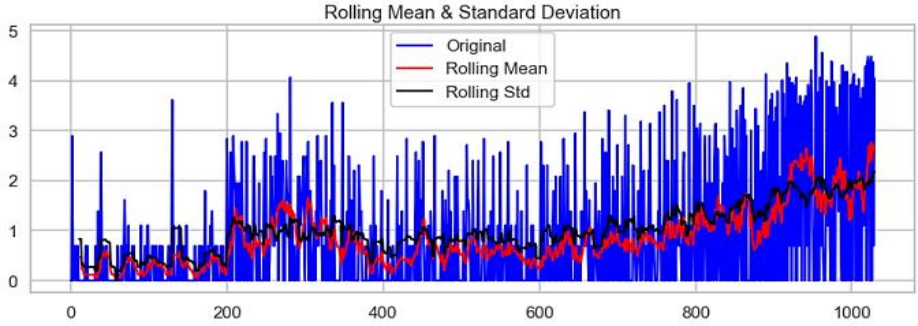

Figure 6. Rolling mean standard deviation (Microsoft)

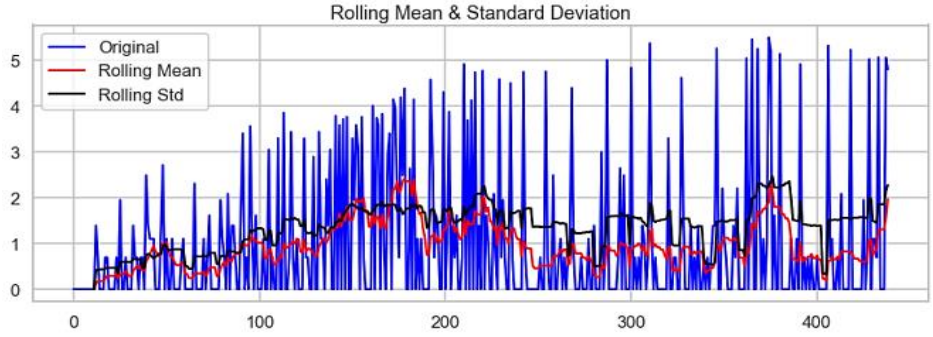

Figure 7. Rolling mean standard deviation (Oracle)

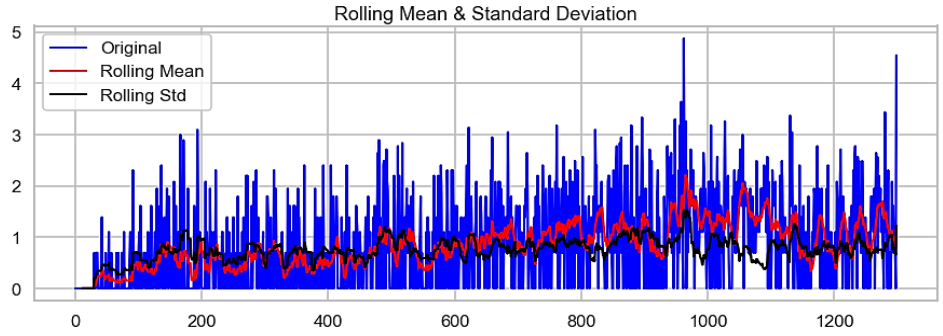

Figure 8. Rolling mean standard deviation (IBM)

```
In [22]: ad_fuller_test(log_series)
Test Statistic                  -1.459632
p-value                          0.553378
#Lags Used                      20.000000
Number of Observations Used    1009.000000
Critical Value (1%)             -3.436848
Critical Value (5%)             -2.864409
Critical Value (10%)            -2.568297
dtype: float64
```

Figure 9. Test statistics value (Microsoft) (snipped from Spyder IDE)

```
ADF Statistic: -8.111298707520387
n_lags: 1.2238206274495662e-12
p-value: 1.2238206274495662e-12
Critial Values:
   1%, -3.4460159927788574
Critial Values:
   5%, -2.868446209372638
Critial Values:
   10%, -2.570448781179138

In [35]:
```

Figure 10. Test statistics value (Oracle) (snipped from Spyder IDE)

```
In [49]: runfile('C:/Users/UNITEN/D
means.py', wdir='C:/Users/UNITEN/De
ADF Statistic: -11.77795816103735
n_lags: 1.0514064047198356e-21
p-value: 1.0514064047198356e-21
Critial Values:
   1%, -3.4354932690454993
Critial Values:
   5%, -2.863811309309343
Critial Values:
   10%, -2.5679792659863123
```

Figure 11. Test statistics value (IBM) (snipped from spyder IDE)

Contrary to feedforward neural networks, the RNNs maintain memory cells to remember the context of the sequence and then use these cells to process the output. The time distributed layer wrapper, provided by keras, is utilized over the dense layer to model the data similar to a time series sequence which applies the same transformation or processing to each time step in the sequence and provides the output after each step. Therefore, it fulfils the requirement of getting the processed output after each input instead of waiting for the whole process to be completed to obtain the output at the end [34]. In this way, the RNN algorithm is applied for the same transformation to every element of the time series where the output values are dependent on the previous values. The LSTM model was created with a parameter optimization for 4 hidden units and 150 epochs. Figures 12-14 show the forecasted plots for different datasets. This proposed model has outperformed the previous models in terms of accuracy. Table 1 shows the results of the accuracy performance (RMSE) of the model, together with the comparison with the previous best performing models [15].

The forecasted plots in Figures 12-14 show a promising picture. The grey, green and blue coloured lines show the true, training fit and testing forecast respectively. It can be observed that the training fit is

nearly perfect for the IBM vendor vulnerability dataset, having achieved the highest accuracy (root mean spurred error value of 0.072). The testing performance or the forecast also showed a decent performance. Although other results involving vulnerability datasets for Microsoft and oracle (RMSE: accuracy values of 0.17 and 0.24 respectively) are not as good as the IBM dataset, they are much better compared to the recent best research in [15] as shown in Table 1.

The performance variation for microsoft and oracle datasets is due to the presence of high spikes in data points in recent time since greater number of vulnerabilities have been discovered in recent years (2016-2019) in the NVD database [35], and in contrary to the absence of such sudden spikes in the past data points; thus, the models had been unable to learn adequately. When more data are added into the national vulnerability database, these models will eventually have the capability to learn adequately for improved performance. Even though the forecast has deviated from the actual at certain places, the overall performance of both in terms of RMSE is promising. Through the use of the time distributed layer wrapper, these models do not only show better performance but also become simpler to reduce computational costs in comparison with the regression approach in some of the previous research. Unlike the previous research, the larger novel dataset of this research has effectively addressed the requirement for the utilization of large training data in obtaining meaningful results from the LSTM network.

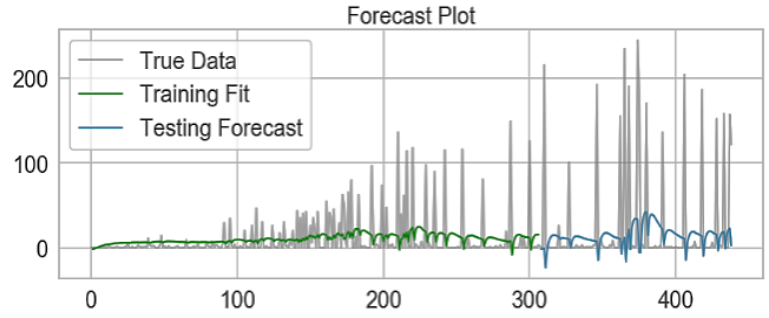

Figure 12. Forecasted plot (Microsoft)

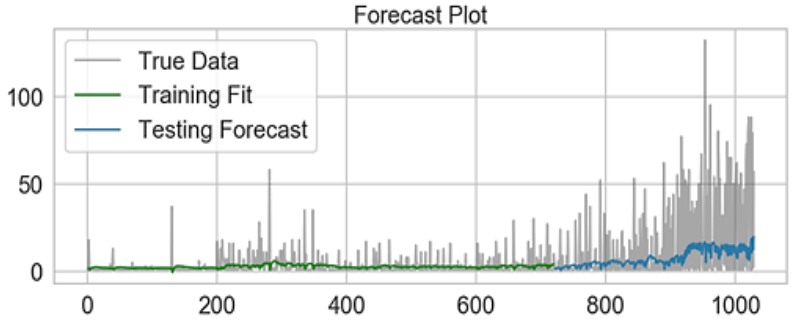

Figure 13. Test statistics value (Oracle)

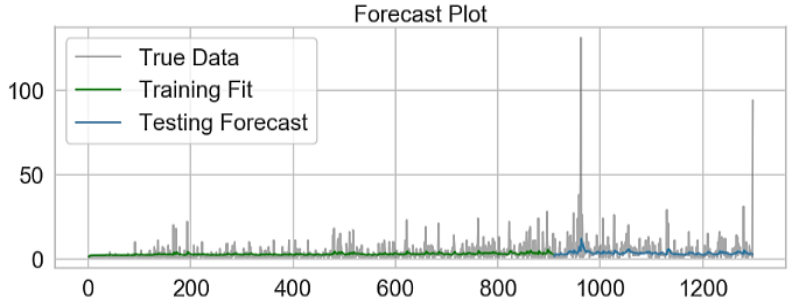

Figure 14. Test statistics value (IBM)

Table 1. Results and comparison

Models of this research (LSTM)		Best previous models proposed by [15]		
Vendor	RMSE	Vendor	Best Model	RMSE
Microsoft	0.17	MAC OS X	ARIMA	19.64
Oracle	0.24	Windows 7	SVM	3.58
IBM	0.072	Linux Kernel	SVM	3.99

6. CONCLUSION

In this research, the ability of the long short-term memory network (LSTM) to learn on its own and use that knowledge to determine the impact of the context and the quantity of past information in forecasting the number of post release vulnerabilities for the top-three vendors namely microsoft, oracle and IBM has been utilized effectively. In the process, we have created a novel dataset for the top-three vendors with the highest number of vulnerabilities in NVD that contain a lot of more historical data points than the previous research in training the LSTM network which prefers a larger dataset in learning to demonstrate better accuracy. To the best of our knowledge, this work is the first to utilize such dataset feature and LSTM deep learning network for this problem in order to develop a vulnerability discovery model (VDM). Our model demonstrated better performance in accuracy (root mean squared error) than other existing such models. The developers, user community, and individual organizations can utilize our model as their respective requirement in managing cyber threats in a cost-effective way. However, an inherent limitation of the LSTM network is that it is not effective to be used with small datasets. Hence, there is always a race in creating even larger datasets than the one utilized in this work for future improvements when necessary. Another limitation is that since the LSTM network uses a lot of loops and memory nodes, it therefore requires longer computation time. However, this drawback can be addressed with the use of powerful machines in the production scenario. Since the long short-term memory (LSTM) recurrent neural network is fully capable of addressing multiple input variables in predicting future values, we therefore plan to extract and generate more novel features from the NVD database in developing a multivariate time series model for this problem as our future work. We also plan to develop a model with 'Published Date' feature as the target label and to integrate it with this current model to add novel forecasting service of vulnerabilities.

ACKNOWLEDGEMENT

Authors acknowledge Universiti Tenaga Nasional (@UNITEN) for its support to the researcherfrom the research grant code of RJO10517844/063 under BOLD program. Appreciations are also extended to ICT Ministry of Bangladesh for its support.

FUNDING

This project was funded by the TNB Seed Fund 2019 Project with a project code 'U-TC-RD-19-09' and ICT Ministry, Bangladesh.

REFERENCES

[1] HP Identifies Top Enterprise Security Threats 2014, [Online]. Available: https://www8.hp.com/us/en/hp-news/press-release.html?id=1571359.
[2] S. Frei, M. May, U. Fiedler, and B. Plattner, "Large-scale Vulnerability Analysis," In *Proceedings of the 2006 SIGCOMM workshop on Large-scale attack defense*, 2006, pp. 131-136, doi: 10.1145/1162666.1162671.
[3] S. Frei, D. Schatzmann, B. Plattner, and B. Trammell, "Modelling the Security Ecosystem-The Dynamics of (In) Security," in *WEIS*, pp. 79-106, 2009, doi: 10.1007/978-1-4419-6967-5_6.
[4] Machine Learning, [Online]. Available: http://www.contrib.andrew.cmu.edu/~mndarwis/ML.html.
[5] A. Mourad, M. A. Laverdiere, and M. Debbabi, "An aspect-oriented approach for the systematic security hardening of code," *Computer and Security*, vol. 27, no. 3-4, pp. 101-114, 2008, doi: 10.1016/j.cose.2008.04.003.
[6] A. Mourad, M. A. Laverdiere, and M. Debbabi, ''High-level aspect-oriented-based framework for software security hardening, *Information Security Journal: A Global Perspective*, vol. 17, no. 2, pp. 56-74, 2008, doi: 10.1080/19393550801911230.
[7] Z. Han, X. Li, Z. Xing, H. Liu, and Z. Feng, "Learning to Predict Severity of Software Vulnerability Using Only Vulnerability Description," *2017 IEEE International Conference on Software Maintenance and Evolution (ICSME)*, Shanghai, China, 2017, pp. 125-136, doi: 10.1109/ICSME.2017.52.
[8] H. Okamura, M. Tokuzane and T. Dohi, "Optimal Security Patch Release Timing under Non-homogeneous Vulnerability-Discovery Processes," *2009 20th International Symposium on Software Reliability Engineering*, Mysuru, India, 2009, pp. 120-128, doi: 10.1109/ISSRE.2009.19.

[9] M. R. Lyu, "Handbook of software reliability engineering," *IEEE Computer Society Press; McGraw Hill*, Los Alamitos, California: New York. 1996.
[10] J. A. Ozment, "*Vulnerability discovery and software security*," 2007. PhD Thesis, University of Cambridge, 2007.
[11] E. Rescorla, "Security holes... Who cares?," In *USENIX Security Symposium*, 2003, p. 75-90.
[12] O. H. Alhazmi and Y. K. Malaiya, "Modeling the vulnerability discovery process," *16th IEEE International Symposium on Software Reliability Engineering (ISSRE'05)*, Chicago, IL, USA, 2005, pp. 1-10, doi: 10.1109/ISSRE.2005.30.
[13] O. H. Alhazmi and Y. K. Malaiya, "Quantitative vulnerability assessment of systems software," *Annual Reliability and Maintainability Symposium, 2005. Proceedings*, Alexandria, VA, USA, 2005, pp. 615-620, doi: 10.1109/RAMS.2005.1408432.
[14] Y. Roumani, J. K. Nwankpa, and Y. F. Roumani, "Time series modeling of vulnerabilities," *Computers and Security*, vol. 51, pp. 32-40, 2015, doi: 10.1016/j.cose.2015.03.003.
[15] N. R. Pokhrel, H. Rodrigo, and C. P. Tsokos, "Cybersecurity: Time Series Predictive Modeling of Vulnerabilities of Desktop Operating System Using Linear and Non Linear Approach," *Journal of Information Security*, vol. 8, no. 04, pp. 362-382, 2017, doi: 10.4236/jis.2017.84023.
[16] R. J. Anderson, "Security in Opens versus Closed Systems-The Dance of Boltzmann, Coase and Moore," *Open Source Software: Economics, Law and Policy*, Toulouse, France, June 20-21, pp. 1-15, 2002.
[17] O. H. Alhazmi and Y. K. Malaiya, "Application of Vulnerability Discovery Models to Major Operating Systems," in *IEEE Transactions on Reliability*, vol. 57, no. 1, pp. 14-22, March 2008, doi: 10.1109/TR.2008.916872.
[18] E. Rescorla, "Is finding security holes a good idea?," in *IEEE Security and Privacy*, vol. 3, no. 1, pp. 14-19, Jan.-Feb. 2005, doi: 10.1109/MSP.2005.17.
[19] J. Kim, Y. K. Malaiya and I. Ray, "Vulnerability Discovery in Multi-Version Software Systems," *10th IEEE High Assurance Systems Engineering Symposium (HASE'07)*, Plano, TX, USA, 2007, pp. 141-148, doi: 10.1109/HASE.2007.55.
[20] P. Li, M. Shaw, and J. Herbsleb, "Selecting a defect prediction model for maintenance resource planning and software insurance," *EDSER-5 Affiliated with ICSE*, 2003, pp. 32-37.
[21] M. Xie, "Software Reliability Modelling, World Scientific," Publishing Co. Pte. Ltd, Singapore, vol. 1, pp. 10-11, 1991.
[22] S. Woo, O. H. Alhazmi, and Y. K. Malaiya, "Assessing Vulnerabilities in Apache and IIS HTTP Servers," *2006 2nd IEEE International Symposium on Dependable, Autonomic and Secure Computing*, Indianapolis, IN, USA, 2006, pp. 103-110, doi: 10.1109/DASC.2006.21.
[23] F. Massacci and V. H. Nguyen, "An Empirical Methodology to Evaluate Vulnerability Discovery Models," in *IEEE Transactions on Software Engineering*, vol. 40, no. 12, pp. 1147-1162, 2014, doi: 10.1109/TSE.2014.2354037.
[24] A. K. Shrivastava, R. Sharma and P. K. Kapur, "Vulnerability discovery model for a software system using stochastic differential equation," *2015 International Conference on Futuristic Trends on Computational Analysis and Knowledge Management (ABLAZE)*, Greater Noida, India, 2015, pp. 199-205, doi: 10.1109/ABLAZE.2015.7154992.
[25] H. C. Joh and Y. K. Malaiya, "Modeling Skewness in Vulnerability Discovery: Modeling Skewness in Vulnerability Discovery," *Quality and Reliability Engineering International*, vol. 30, no. 8, pp. 1445-1459, 2014, doi: https://doi.org/10.1002/qre.1567.
[26] A. A. Younis, H. Joh, and Y. Malaiya, "Modeling Learningless Vulnerability Discovery using a Folded Distribution," In: *Proceedings of the International Conference on Security and Management (SAM)*, 2011, pp. 617-623.
[27] O. H. Alhazmi and Y. K. Malaiya., "Measuring and Enhancing Prediction Capabilities of Vulnerability Discovery Models for Apache and IIS HTTP Servers," *2006 17th International Symposium on Software Reliability Engineering*, Raleigh, NC, USA, 2006, pp. 343-352, doi: 10.1109/ISSRE.2006.26.
[28] L. Wang, Y. Zeng, and T. Chen, "Back propagation neural network with adaptive differential evolution algorithm for time series forecasting," *Expert Systems with Applications*, vol. 42, no. 2, pp. 855-863, 2015, doi: 10.1016/j.eswa.2014.08.018.
[29] Y. Movahedi, M. Cukier, A. Andongabo, and I. Gashi, "Cluster-Based Vulnerability Assessment Applied to Operating Systems," *2017 13th European Dependable Computing Conference (EDCC)*, Geneva, Switzerland, 2017, pp. 18-25, doi: 10.1109/EDCC.2017.27.
[30] Top 50 Vendors by Total Number of, "Distinct," *Vulnerabilities*, [Online]. Available: https://www.cvedetails.com/top-50-vendors.php.
[31] Understanding LSTM Networks, [Online]. Available: http://colah.github.io/posts/2015-08-Understanding-LSTMs/.
[32] How to work with Time Distributed data in a neural, network, [Online]. Available: https://medium.com/smileinnovation/how-to-work-with-time-distributed-data-in-a-neural-network-b8b39aa4ce00.
[33] LSTM Forward and Backward Pass, Introduction**,** [Online]. Available: http://arunmallya.github.io/writeups/nn/lstm/index.html#/.
[34] A Brief Summary of the Math Behind RNN (Recurrent Neural Networks), [Online]. Available: https://medium.com/towards-artificial-intelligence/a-brief-summary-of-maths-behind-rnn-recurrent-neural-networks-b71bbc183ff.
[35] NIST, National Vulnerability Database, [Online]. Available: https://nvd.nist.gov/.

BIOGRAPHIES OF AUTHORS

Mohammad Shamsul Hoque holds a Master's degree in information security and digital forensics from the University of East London, U.K. He is currently a PhD student at CCI, Universiti Tenaga Nasional, Malaysia. His research focus is cyber security and artificial intelligence of critical system and facilities.

Norziana Binti Jamil received her PhD (Security in Computing) from Universiti Putra Malaysia in 2013. She is currently an Associate Professor at the College of Computing and Informatics, Universiti Tenaga Nasional, Malaysia. Her area of specialization and research interests are cryptography, security of cyber physical systems, privacy preserving techniques and security intelligence and analytics.

Nowshad Amin received his PhD in Electrical and Electronics Engineering from Tokyo Institute of Technology, Japan. He is a Professor in Renewable Energy and Solar Photovoltaics, Institute of Sustainable Energy (ISE), Universiti Tenaga Nasional, Malaysia. His research interests are micro & nano-electronics, solar photovoltaic energy applications, smart controllers, HEMS and cyber security in power plant system.

Azril Azam Abdul Rahim has currently joined the ranks of a Senior Manager with Tenaga Nasional Berhad (TNB). He currently manages the Cyber Threat Intelligence Management for TNB. To date, he has more than 20 years of experience in cyber security. Graduated with a double degree in computer science and business operation from the University of Missouri, Azril also holds various certifications such as GCFA, ECSP, CEH and CEI.

Razali B. Jidin past research activities were on embedded system (ES), focusing implementation of *multi-threading semaphores* into Field Programmable Gate Arrays (*FPGA*), targeted for real-time systems. Recent activities are on architectures and mathematical transforms of *lightweight cryptography* for implementation on resource-constraint devices. His other area of research activities was related to renewable energy with projects such as finding an optimal combination of diesel and renewable energies for a resort island, photovoltaic and battery storages for underserved communities and application Internet of things (IoT). Dr. Razali obtained his Ph.D. in Electrical Engineering & Computer Science, Univ. of Kansas, Lawrence, KS, USA. He is registered with Board of Engineer Malaysia (BEM) as Professional Electrical Engineer, and with Institute Engineer Malaysia in area of Control & Instrumentation.