A Project report on

Application of Robust Software Modelling Tool for Web Attacks Detection

A Dissertation submitted to JNTU Hyderabad in partial fulfillment of the
academic requirements for the award of the degree.

Bachelor of Technology

in

Computer Science and Engineering

Submitted by

BALLA GANESH
(20H51A05B1)

LANKA SHRIYA
(20H51A05E5)

KALLURI RISHITA
(20H51A0535)

Under the esteemed guidance of

Mr. A. Vivekanand
(Assistant Professor)

EXPLORE TO INVENT

Department of Computer Science and Engineering

CMR COLLEGE OF ENGINEERING& TECHNOLOGY

(UGC Autonomous)
*Approved by AICTE *Affiliated to JNTUH *NAAC Accredited with A^{+} Grade

KANDLAKOYA, MEDCHAL ROAD, HYDERABAD - 501401.

2020- 2024

CMR COLLEGE OF ENGINEERING & TECHNOLOGY

KANDLAKOYA, MEDCHAL ROAD, HYDERABAD – 501401

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the Major Project Phase I report entitled **"Application of Robust Software Modelling Tool for Web Attacks Detection"** being submitted by Balla Ganesh (20H51A05B1), Lanka Shriya (20H51A05E5), Kalluri Rishita (20H51A0535) in partial fulfillment for the award of **Bachelor of Technology in Computer Science and Engineering** is a record of bonafide work carried out his/her under my guidance and supervision.

The results embodies in this project report have not been submitted to any other University or Institute for the award of any Degree.

Mr. A. Vivekanand
Assistant Professor
Dept. of CSE

Dr. Siva Skandha Sanagala
Associate Professor and HOD
Dept. of CSE

ACKNOWLEDGEMENT

With great pleasure we want to take this opportunity to express my heartfelt gratitude to all the people who helped in making this project work a grand success.

We are grateful to **Mr. A. Vivekanand, Assistant Professor** , Department of Computer Science and Engineering for his valuable technical suggestions and guidance during the execution of this project work.

We would like to thank **Dr. Siva Skandha Sanagala,** Head of the Department of Computer Science and Engineering, CMR College of Engineering and Technology, who is the major driving forces to complete my project work successfully.

We are very grateful to **Dr. Vijaya Kumar Koppula**, Dean-Academics, CMR College of Engineering and Technology, for his constant support and motivation in carrying out the project work successfully.

We are highly indebted to **Major Dr. V A Narayana,** Principal, CMR College of Engineering and Technology, for giving permission to carry out this project in a successful and fruitful way.

We would like to thank the **Teaching & Non- teaching** staff of Department of Computer Science and Engineering for their co-operation

We express our sincere thanks to **Shri. Ch. Gopal Reddy**, Secretary, CMR Group of Institutions, for his continuous care.

Finally, We extend thanks to our parents who stood behind us at different stages of this Project. We sincerely acknowledge and thank all those who gave support directly and indirectly in completion of this project work.

Balla Ganesh	20H51A05B1
Lanka Shriya	20H51A05E5
Kalluri Rishita	20H51A0535

TABLE OF CONTENTS

CHAPTER NO.	**TITLE**	**PAGE NO.**
	LIST OF FIGURES	ii
	LIST OF TABLES	iii
	ABSTRACT	iv
1	**INTRODUCTION**	1
	1.1 Problem Statement	2
	1.2 Research Objective	2
	1.3 Project Scope and Limitations	3
2	**BACKGROUND WORK**	**4-10**
	2.1. A classification of sql-injection attacks and countermeasures	5
	2.1.1.Introduction	5
	2.1.2.Merits,Demerits and Challenges	5
	2.1.3.Implementation	6
	2.2. An adaptive network intrusion detection method based on PCA and support vector machines. Advanced Data Mining and Applications	7
	2.2.1.Introduction	7
	2.2.2.Merits,Demerits and Challenges	7
	2.2.3.Implementation	8
	2.3. Using Adaptive Alert Classification to Reduce False Positives in Intrusion Detection	9
	2.3.1.Introduction	9
	2.3.2.Merits,Demerits and Challenges	9
	2.3.3.Implementation	10
3	**RESULTS AND DISCUSSION**	**11-13**
	4.1. Performance Metrics	11-13
4	**CONCLUSION**	14-15
	5.1 Conclusion	15
	REFERENCES	**16-18**

List of Figures

FIGURE NO.	TITLE	PAGE NO.
1.1	Sql Injection attack system	6
2.1	Intrusion detection system	8
3.1	Adaptive alert classification	10

List of Tables

FIGURE NO.	TITLE	PAGE NO.
1.1	Table for sql Injection Attack .	12
2.1	False Positive Reduction Technique	13

ABSTRACT

Web applications are popular targets for cyber-attacks because they are network-accessible and often contain vulnerabilities. An intrusion detection system monitors web applications and issues alerts when an attack attempt is detected. Existing implementations of intrusion detection systems usually extract features from network packets or string characteristics of input that are manually selected as relevant to attack analysis. Manually selecting features, however, is time-consuming and requires in-depth security domain knowledge. Moreover, large amounts of labeled legitimate and attack request data are needed by supervised learning algorithms to classify normal and abnormal behaviors, which is often expensive and impractical to obtain for production web applications. This paper provides three contributions to the study of autonomic intrusion detection systems. First, we evaluate the feasibility of an unsupervised/semi-supervised approach for web attack detection based on the Robust Software Modeling Tool (RSMT), which autonomically monitors and characterizes the runtime behavior of web applications. Second, we describe how RSMT trains a stacked denoising autoencoder to encode and reconstruct the call graph for end-to-end deep learning, where a low-dimensional representation of the raw features with unlabeled request data is used to recognize anomalies by computing the reconstruction error of the request data. Third, we analyze the results of empirically testing RSMT on both synthetic datasets and production applications with intentional vulnerabilities. Our results show that the proposed approach can efficiently and accurately detect attacks, including SQL injection, cross-site scripting, and deserialization, with minimal domain knowledge and little labeled training data.

CHAPTER 1
INTRODUCTION

CHAPTER 1

INTRODUCTION

1.1.Problem Statement

Web applications are prime targets for cyber-attacks due to their network accessibility and potential vulnerabilities. Existing intrusion detection systems often rely on manually selected features or require large labeled datasets for supervised learning, making them time-consuming and costly. This research aims to address these limitations by evaluating an unsupervised/semi-supervised approach based on the Robust Software Modeling Tool (RSMT) to autonomically monitor web application behavior, train deep learning models, and efficiently detect attacks like SQL injection and cross-site scripting with minimal domain knowledge and labeled data.

1.2.Research Objective

The objective of this study is threefold. Firstly, to assess the feasibility of employing an unsupervised/semi-supervised approach for web attack detection using RSMT, which characterizes web application runtime behavior. Second, to describe how RSMT trains a stacked denoising autoencoder to encode and reconstruct call graphs for end-to-end deep learning, allowing the recognition of anomalies in request data with minimal labeled data. Lastly, to empirically test the RSMT approach on synthetic datasets and real web applications to demonstrate its efficient and accurate detection of various attacks, requiring little domain expertise.

Scope: This research explores an unsupervised/semi-supervised intrusion detection approach using RSMT, emphasizing efficient detection of web application attacks, including SQL injection, cross-site scripting, and deserialization. Limitations: Challenges may arise in adapting to emerging attack methods, and the real-world applicability and effectiveness of RSMT in various web application settings may need further assessment.

1.3.Project Scope and Limitations

- This project explores an unsupervised/semi-supervised intrusion detection approach using RSMT, emphasizing efficient detection of web application attacks, including SQL injection, cross-site scripting, and deserialization.

- We are using LSTM (Long Short Term Memory) Algorithm which is an advance version of deep learning network whose prediction accuracy is more compare to existing algorithms.
- Automatically detect attacks on web applications

Limitations

- Address limitations regarding the availability, diversity, and quality of the sequential web traffic data. Insufficient or biased data might impact the RSMT model's ability to generalize well to real-world scenarios.

- Acknowledge the complexity of the RSMT architecture. Complex models often require significant computational resources and longer training times. Discuss potential challenges related to computational limitations.

CHAPTER 2
BACKGROUND WORK

CHAPTER 2

BACKGROUND WORK

2.1 A classification of sql-injection attacks and countermeasures:

2.1.1 Introduction:

SQL injection attacks are a significant threat to the security of web applications, allowing attackers to gain unauthorized access to databases and sensitive information. Current approaches to mitigating these attacks have limitations or do not encompass the full spectrum of SQL injection vulnerabilities. Research paper addresses this issue by offering an extensive review of various SQL injection attack types, complete with descriptions and examples. It also examines existing detection and prevention techniques, highlighting their strengths and weaknesses in tackling the wide range of SQL injection attacks.

2.1.2 Merits, Demerits and Challenges:

Merits:

1. The classification provides a detailed understanding of different SQL injection attack types, aiding researchers and practitioners in recognizing vulnerabilities effectively.

2. By offering real-world examples for each attack type, it enhances comprehension and assists in the development of countermeasures.

Demerits:

1. The classification does not prioritize attacks, making it challenging to focus efforts on the most critical vulnerabilities.

2. As the paper reviews existing techniques, it might not cover the most recent and evolving SQL injection tactics and countermeasures.

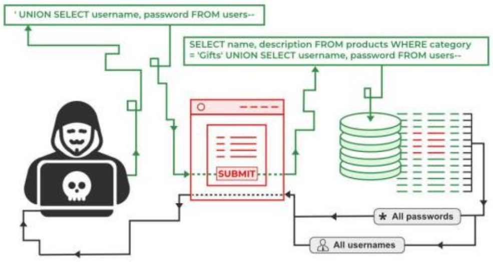
Challenges:

1. As SQL injection tactics evolve, staying up to date and adaptable to new attack vectors is a persistent challenge.

2. Implementing effective prevention measures may require significant resources, posing a challenge for resource-constrained organizations.

2.1.3 Implementation:

Implementing the findings of this research involves using the comprehensive classification of SQL injection attacks to develop and refine detection and prevention techniques. Security practitioners can apply the knowledge gained to bolster their security measures, incorporating real-world attack examples to enhance their defense strategies. The evaluation of existing techniques helps in selecting and fine-tuning the most suitable countermeasures. This research serves as a valuable resource for organizations striving to secure their web applications against SQL injection threats.



```
' UNION SELECT username, password FROM users--
```

```
SELECT name, description FROM products WHERE category
= 'Gifts' UNION SELECT username, password FROM users--
```

Figure.1.1: Sql injection attack system

2.2 An adaptive network intrusion detection method based on pca and support vector machines. Advanced Data Mining and Applications:

2.2.1 Introduction:

Network intrusion detection is crucial for computer security, but existing intrusion detection systems (IDSs) face challenges in keeping up with the evolving threat landscape and increasing network traffic speed. This paper presents an innovative adaptive intrusion detection method that combines Principal Component Analysis (PCA) and Support Vector Machines (SVMs). By leveraging PCA for dimension reduction and SVMs for classification, this method aims to enhance the accuracy and speed of IDSs, with the potential to handle new attacks effectively.

2.2.2 Merits, Demerits and Challenges:

Merits:

1. The method offers faster training and detection speeds, addressing the need for real-time intrusion detection in high-traffic networks.

2. SVMs are known for their robust generalization capabilities, contributing to strong classification performance.

Demerits:

1. The computational resources needed for SVMs and dimension reduction might be substantial, affecting the feasibility for resource-constrained environments.

2. The effectiveness of PCA and SVMs may be impacted by the quality and representativeness of the training data, making data preprocessing critical.

Challenges:

1. Real-world network data can be highly variable, and the model's robustness to these variations remains a challenge in practical deployment.

2. The system's ability to adapt to emerging attack techniques and unknown threats is a persistent challenge in network intrusion detection.

2.2.3 Implementation:

The proposed adaptive intrusion detection method combines PCA and SVMs to reduce dimensionality and enhance classification accuracy and speed. Practitioners can implement this method by integrating PCA into the preprocessing stage of network data, followed by training multi-class SVMs on the transformed data. This approach reduces the need for intricate parameter tuning and offers a faster and more accurate intrusion detection process. The method's effectiveness is supported by experimental results on KDD-Cup99 intrusion detection data, demonstrating its potential for improving network security.

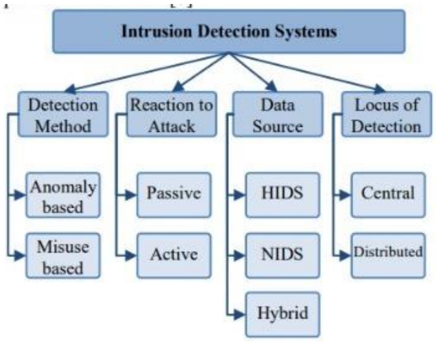


Figure.2.1: Intrusion detection method

2.3 Using Adaptive Alert Classification to Reduce False Positives in Intrusion Detection:

2.3.1 Introduction:

Intrusion Detection Systems (IDSs) play a vital role in computer security, but they often inundate human analysts with numerous false positive alerts. These alerts, triggered by benign events, obscure the identification of true security threats. To address this issue, this paper introduces ALAC (Adaptive Learner for Alert Classification), a novel system designed to reduce false positives in intrusion detection. ALAC dynamically learns from human analysts to classify alerts more accurately and can even autonomously process highly confident alerts, thereby alleviating the analyst's workload.

2.3.2 Merits, Demerits and Challenges:

Merits:

1. The system's ability to autonomously handle confidently classified alerts decreases the analyst's workload, enabling more efficient threat detection.

2. ALAC employs adaptive learning to classify alerts, improving its accuracy over time based on the observations and feedback from human analysts.

Demerits:

1. The adaptive learning process may require initial training and might not be effective until sufficient analyst feedback is accumulated.

2. There is a risk that ALAC could misclassify alerts, either initially or over time, leading to missed true positives or excessive discarding of alerts.

Challenges:

1. Handling a large volume of alerts and accommodating the needs of analysts in high-traffic network environments may be a scalability challenge.

2. Choosing and fine-tuning the appropriate machine learning techniques for adaptive learning can be complex and time-consuming.

2.3.3 Implementation:

Implementation of ALAC involves creating a system that adapts its alert classification based on observations and feedback from human analysts. A prototype of ALAC is developed, and a suitable machine learning technique is chosen to facilitate adaptive learning. The system is integrated into the existing intrusion detection infrastructure, offering the capability to autonomously process alerts classified with high confidence, thereby reducing the analyst's workload. Experimental validation is conducted to assess its effectiveness in facilitating the analyst's work and improving alert accuracy.

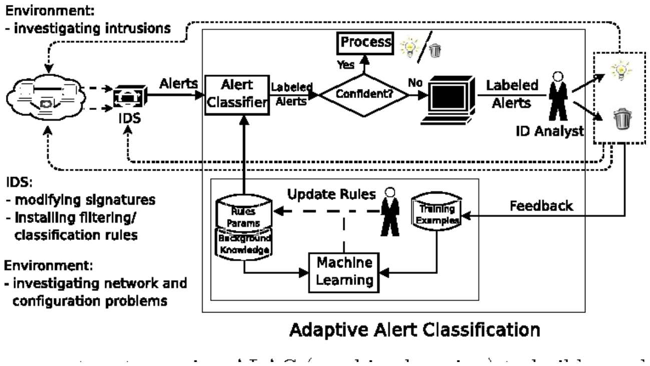
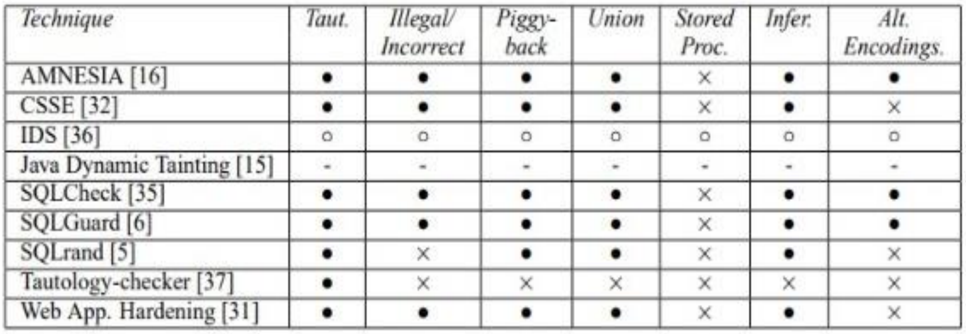


Figure.3.1: Adaptive alert classification

CHAPTER 3
RESULTS AND DISCUSSION

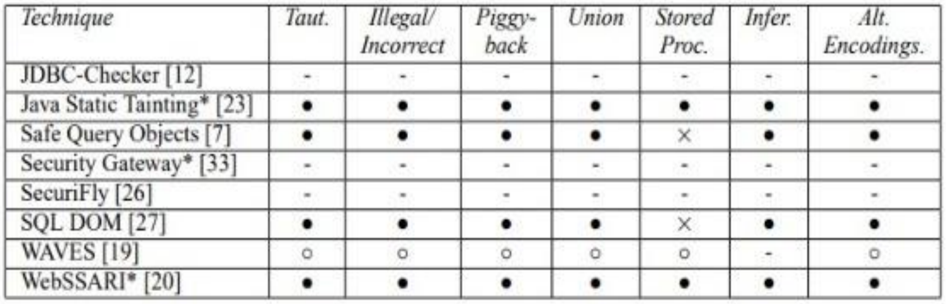
CHAPTER 3

RESULTS AND DISCUSSION



Technique	*Taut.*	*Illegal/ Incorrect*	*Piggy-back*	*Union*	*Stored Proc.*	*Infer.*	*Alt. Encodings.*
AMNESIA [16]	●	●	●	●	×	●	●
CSSE [32]	●	●	●	●	×	●	×
IDS [36]	○	○	○	○	○	○	○
Java Dynamic Tainting [15]	-	-	-	-	-	-	-
SQLCheck [35]	●	●	●	●	×	●	●
SQLGuard [6]	●	●	●	●	×	●	●
SQLrand [5]	●	×	●	●	×	●	×
Tautology-checker [37]	●	×	×	×	×	×	×
Web App. Hardening [31]	●	●	●	●	×	●	×

Table 1: Comparison of detection-focused techniques with respect to attack types.



Technique	*Taut.*	*Illegal/ Incorrect*	*Piggy-back*	*Union*	*Stored Proc.*	*Infer.*	*Alt. Encodings.*
JDBC-Checker [12]	-	-	-	-	-	-	-
Java Static Tainting* [23]	●	●	●	●	●	●	●
Safe Query Objects [7]	●	●	●	●	×	●	●
Security Gateway* [33]	-	-	-	-	-	-	-
SecuriFly [26]	-	-	-	-	-	-	-
SQL DOM [27]	●	●	●	●	×	●	●
WAVES [19]	○	○	○	○	○	-	○
WebSSARI* [20]	●	●	●	●	●	●	●

Table 2: Comparison of prevention-focused techniques with respect to attack types.

Table.1.1: Comparison Matrix table for sql Injection Attack

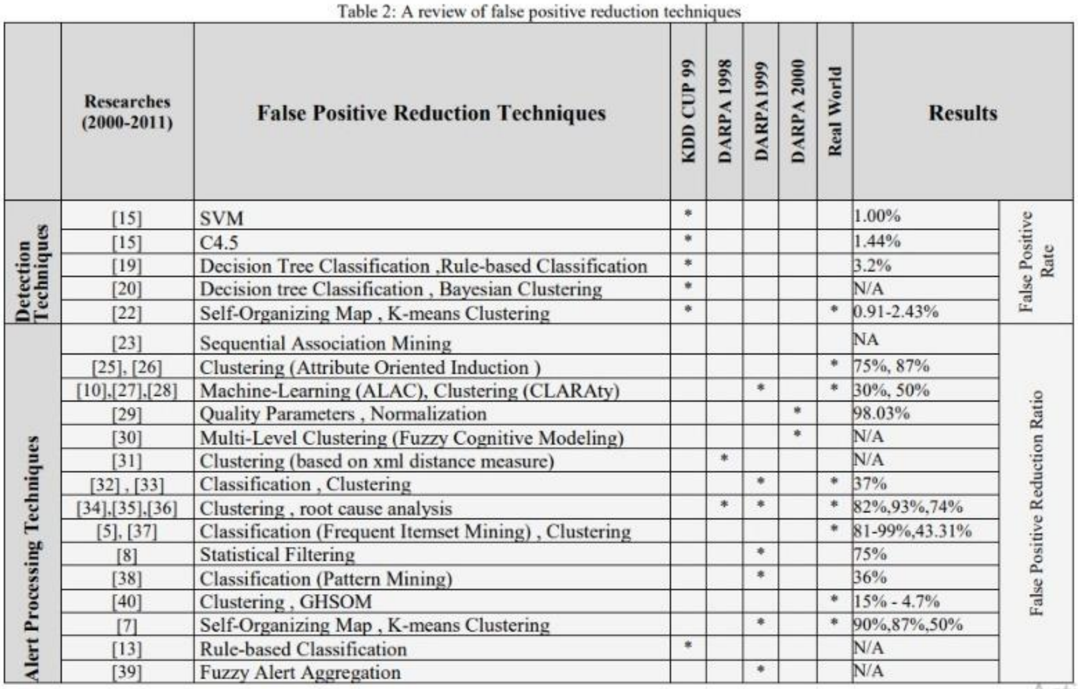


Table 2: A review of false positive reduction techniques

	Researches (2000-2011)	False Positive Reduction Techniques	KDD CUP 99	DARPA 1998	DARPA1999	DARPA 2000	Real World	Results	
Detection Techniques	[15]	SVM	*					1.00%	False Positive Rate
	[15]	C4.5	*					1.44%	
	[19]	Decision Tree Classification ,Rule-based Classification	*					3.2%	
	[20]	Decision tree Classification , Bayesian Clustering	*					N/A	
	[22]	Self-Organizing Map , K-means Clustering	*				*	0.91-2.43%	
Alert Processing Techniques	[23]	Sequential Association Mining						NA	False Positive Reduction Ratio
	[25], [26]	Clustering (Attribute Oriented Induction)					*	75%, 87%	
	[10],[27],[28]	Machine-Learning (ALAC), Clustering (CLARAty)			*		*	30%, 50%	
	[29]	Quality Parameters , Normalization				*		98.03%	
	[30]	Multi-Level Clustering (Fuzzy Cognitive Modeling)				*		N/A	
	[31]	Clustering (based on xml distance measure)		*				N/A	
	[32] , [33]	Classification , Clustering			*		*	37%	
	[34],[35],[36]	Clustering , root cause analysis		*	*		*	82%,93%,74%	
	[5], [37]	Classification (Frequent Itemset Mining) , Clustering					*	81-99%,43.31%	
	[8]	Statistical Filtering			*			75%	
	[38]	Classification (Pattern Mining)			*			36%	
	[40]	Clustering , GHSOM					*	15% - 4.7%	
	[7]	Self-Organizing Map , K-means Clustering			*		*	90%,87%,50%	
	[13]	Rule-based Classification	*					N/A	
	[39]	Fuzzy Alert Aggregation			*			N/A	

Table.2.1: Comparison Matrix table for False Positive Reduction techniques

CHAPTER 4
CONCLUSION

CHAPTER 4

CONCLUSION

This project describes the architecture and results of applying a unsupervised end-to-end deep learning approach to automatically detect attacks on web applications. We instrumented and analyzed web applications using the Robust Software Modeling Tool (RSMT), which autonomically monitors and characterizes the runtime behavior of web applications. We then applied a denoising autoencoder to learn a low-dimensional representation of the call traces extracted from application runtime. To validate our intrusion detection system, we created several test applications and synthetic trace datasets and then evaluated the performance of unsupervised learning against these datasets. While cross validation is widely used in traditional machine learning, it is often not used for evaluating deep learning models because of the great computational cost.

REFERENCES

REFERENCES

1. Halfond WG, Viegas J, Orso A. A classification of sql-injection attacks and countermeasures. In: Proceedings of the IEEE International Symposium on Secure Software Engineering. IEEE; 2006. p. 13–5.

2. Wassermann G, Su Z. Static detection of cross-site scripting vulnerabilities. In: Proceedings of the 30th International Conference on Software Engineering. ACM; 2008. p. 171–80.

3. Di Pietro R, Mancini LV. Intrusion Detection Systems vol. 38: Springer; 2008.

4. Qie X, Pang R, Peterson L. Defensive programming: Using an annotation toolkit to build dos-resistant software. ACM SIGOPS Oper Syst Rev. 2002;36(SI):45–60.

5. https://doi.org/https://www.acunetix.com/acunetix-web-applicationvulnerability-report-2016. Accessed 16 Aug 2017.

6. https://doi.org/http://money.cnn.com/2015/10/08/technology/ cybercrime-cost-business/index.html. Accessed 16 Aug 2017.

7. https://doi.org/https://www.consumer.ftc.gov/blog/2017/09/equifaxdata-breach-what-do. Accessed 16-August-2017.

8. https://doi.org/https://theconversation.com/why-dont-big-companieskeep-their-computer-systems-up-to-date-84250. Accessed 16 Aug 2017.

9. Ben-Asher N, Gonzalez C. Effects of cyber security knowledge on attack detection. Comput Hum Behav. 2015;48:51–61.

10. Japkowicz N, Stephen S. The class imbalance problem: A systematic study. Intell Data Anal. 2002;6(5):429–49.

11. Liu G, Yi Z, Yang S. A hierarchical intrusion detection model based on the pca neural networks. Neurocomputing. 2007;70(7):1561–8.

12. Xu X, Wang X. An adaptive network intrusion detection method based on pca and support vector machines. Advanced Data Mining and Applications. 2005;3584:696–703.

13. Pietraszek T. Using adaptive alert classification to reduce false positives in intrusion detection. In: Recent Advances in Intrusion Detection. Springer; 2004. p. 102–24.

14. Goodfellow I, Bengio Y, Courville A. Deep Learning: MIT press; 2016.

15. Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems. Curran Associates, Inc.; 2012. p. 1097–105.