ATTACKS ON WEB LOG DATA: A REVIEW

Diwakar Prasad Nuniya[1], Nisha[2]

[1]Research Scholar, [2]Assistant Professor
Department of Computer Science,
PKG Group of Institutions, Panipat

***Abstract*: Many methods have been developed to protect web servers against attacks. Anomaly detection methods rely on generic user models and application behaviour, which interpret departures as indications of potentially dangerous behavior from the established pattern. In this paper, we conducted the use of a systematic review of the anomaly detection methods to prevent and identify web assaults; Many techniques of anomaly identification for automated log analysis have been suggested to minimize manual work. However, due to a lack of evaluations and comparisons of various anomaly detection techniques, engineers may still decide which detection methods should not be used. Furthermore, even if engineers use an unusual detection technique, re-implementation will take a lifetime. We offer a comprehensive analysis and evaluation of six existing log-based detection techniques, including three monitored and three unchecked modes, as well as an open toolkit that allows for simple reuse, to address these problems. These techniques were evaluated on two production log databases produced by the public, with a total of 15,923,592 log messages and 365,298 anomaly cases. We think that our work, as well as the testing results and associated discoveries, may be used as guidelines for adopting these strategies and as a source of inspiration for future research.**

***Keywords*: Anomaly Detection; Web Attacks; Log Anomaly, Web Content**

1. INTRODUCTION

Web mining is the process of generation of human readable information from web log data. The website's server is the primary source for web log data.. Web mining converts the server web log data into easily understandable information. This systematic human readable information is very useful for security, improvement and maintenance purposes of websites.

Web mining uses nearly all mathematical models of data mining. So, web mining may be defined as the branch of data mining in which web server logs are used as data. Web mining is the analysis of web server log data for the sake of generation of valuable information. Analysis of server log data gives marvelous insights about visitor's navigational behavior, development of personalization systems, accessing website security, target marketing, improvement/development of websites, improving server's performance and many more. Web logs contain structured and unstructured data. Depending on the data type, web mining is further categorized in three parts [Figure 1.1].

- ❖ Web-Content-Mining
- ❖ Web-Structure-Mining
- ❖ Web-Log/Usage-Mining

Web-Content-Mining

This mining is related to the generation of useful patterns from elements of a website like images, tables, text, audio, video, pdf, graphics etc. There are two approaches to web content mining. First one related to the mining of the contents of web pages. It gives mining results based on the type of content. Second approach based on the target of mining or improving web search results of search engines. The classification of websites based on contents on web pages is an example of the second approach of web content mining.

Web Structure Mining

Web structure mining deals with the hyperlink connectivity within the webpage and in between the webpages. Web Structure Mining explains the hyperlink structure of a website to obtain the insight for improvement of the website. Websites link structure analyzed in web structure mining. The relationship between websites can be analyzed by organizing link structure in the form of topology. Web page ranking and website reorganization are two direct applications of web structure mining.

Web Log/Usage Mining

The analysis of web log data for discovery of navigational patterns and analysis of these discovered patterns for generation of standard rules comes under web log mining. Web usage mining refers to extraction of valuable usage patterns from the analysis of web log data. Thus 'web log mining' and 'web usage mining' represent the same meaning. Web usage mining is the analysis of web log data for generation of standard rules and usage patterns.

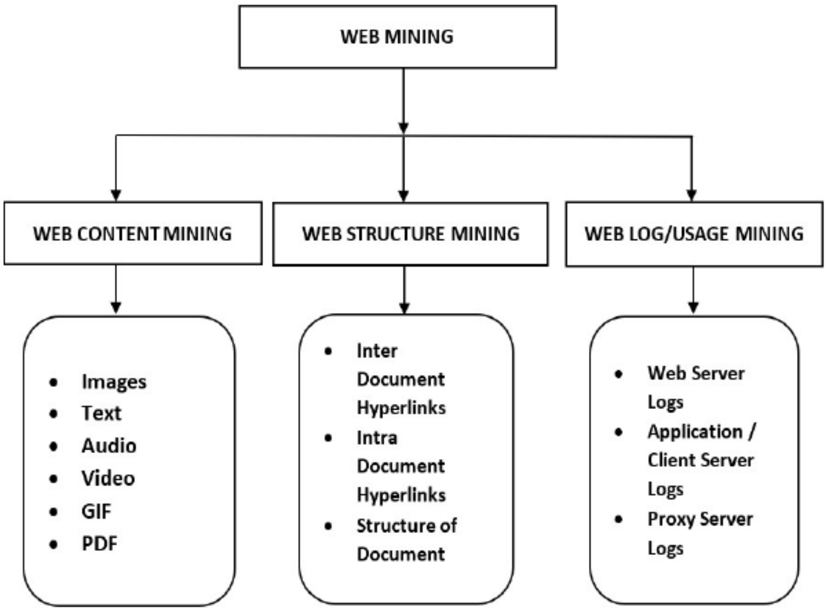

Figure 1.1: Types of Web mining.

In web logs each clickstream of a visitor's traversal is recorded. So, web log data is an immense source of information about visitors of websites. It contains information of visitors like IP address, user's identification details, date and time of visit, bytes transferred, status code of request, user's system information (user agent), resource requested, referrer websites, protocol used for request, Cookie etc. Analysis of this information is vital for web administrators for website development, website improvement, personalization, security and other visitors related issues.

Srivastava J et al described the complete web mining process for mining web log data, giving special attention to commercial applications [1]. Agarwal et al initially presented a most researched pattern discovery association rule mining algorithm [2]. Cooley R et al discussed different pattern discovery methods for web log data [3]. Researchers [4] given detailed analysis of pre-processing methods for web usage mining. They provided the WEBMINER system for the complete web mining process. WEBMINER incorporated pre-processing, knowledge discovery and pattern analysis phases in the system architecture. According to Cooley et al [5] if a unique IP accesses a web page without using a hyperlink, then the user will be counted as a new user. Martin Arlitt and Carey Williamson used the data from server logs of NASA that was collected by Jim Dumoulin of the Kennedy Space Center [9]. They explained the various features of workloads of the web server.

II. Related Work

The logs were analysed. Log analysis has been used to increase software system dependability[35] in a variety of ways, including anomaly detection[10],[28],[47], failure diagnosis[17],[31],[38], programme verification[11],[42], and act prediction[16]. The majority of these log analysis approaches are divided into two steps: log parsing and log mining, both of which have received a lot of attention in current years. He et al.[24] compare the efficiency of four non-system source code offline log parsing methods: SLCT[45], IPLOM[29], LogSig[44], and LKE[20]. [34] proposes an offline log parsing solution which requires linear time and space. Using system sources, Xu et al.[47] offer an online log processing method. Xu et al[47] employ PCA to find abnormalities, with the input being a matrix built from logs.

Beschastnikh et al.[11] create a finite state machine that defines system runtime behaviour using system logs. Unlike these articles, which use log analysis to resolve a range of complications, we focus on log-based anomaly detection methods.

Anomalies Detection

Anomaly detection is the process of looking for out-of-the-ordinary behaviour that can be stated to manual examination and debugging engineers. Bovenzi et al.[13] present an operating system-level method for detecting abnormalities that is suited for mission-critical systems. Venkatakrishnan et.al[46] identify safety vulnerabilities before a system is compromised.

In contrast to past efforts that concentrated on discovering individual anomalies, this study analyses the efficiency of anomaly detection strategies for generic irregularities in large-scale systems. Babenko et al.[9] offer an algorithm for automatically creating explanations from anomaly-detected failures.

Empirical research

Since empirical research may often provide practical insights to both academics and developers, there has been a lot of empirical research on software dependability in recent years. Yuan et al.[48] investigate open-source logging practises and offer advice to developers.

Fu et al.[21],[49] investigate the logging industry empirically. Pecchia and colleagues [37] look into the goals and difficulties of logging in industrial settings. The use of decision tree approaches to detect smells in code is investigated by Amorim and colleagues [7]. Lanzaro and his colleagues [25] look on how library code flaws emerge as interface issues. [40] Take a look at long-living bugs from five different angles. Milenkoski and colleagues[33] investigate and organise typical computer intrusion detection approaches. Take, for example, Chandola. [14] Survey anomaly detection methods that employ machine learning practices in a range of domains, but this research focuses on assessing and evaluating existing work that employs log analysis to discover system anomalies.

Review of Log Anomalies and Deep Learning

To identify suspect business-specific activity and user profile behaviour, T.F. Yen et al. [29] used SIEM log data composed from over 1.4 billion logs each day. Scalability, data noise, and a lack of ground truth were all challenges for this project. The suggested solution demands the generation of a feature vector based on historical data for each internet host. To detect potential security problems, they utilise unsupervised clustering using data-specific characteristics. Manual labelling experts must be aware of the absence of ground-based reality. The technique is rule-based, and historical log processing requires subject-matter expertise. Min Du et al. [2] proposed an architecture for detecting anomalies in log data that does not need any former knowledge of the domain. The proposed method includes a process for diagnosing log key and parameter value abnormalities, as well as a mechanism for identifying log key and parameter value abnormalities from logs. The probability of the next log key is predicted using a neural network-based method.

A log parameter sequence abnormality can similarly be detected using a comparable LSTM neural network. The software also uses false-positive manual feedback to improve future accuracy. The LSTM considers the log series to be a natural language sequence that may be processed accordingly. Using datasets from BGL, Thunderbird, Open Stack, and IMDB, Amir Farzad et al. [6] suggested a deep learning model for detecting log message abnormalities and compared these models to boost efficiency. The IMDB dataset is used to demonstrate how their method can be used to a range of classification challenges.

Natural Language Processing techniques were used by Mengying Wang et al. [1] to discover abnormal log messages. In the research, word2vec and TF-IDF feature extraction methods are applied, and the activity is finished with a classification LSTM deep learning algorithm. They discovered that word2vec beats TF-IDF in log message identification jobs.

W Meng et al. 2019 [4] created an attention-based LSTM model that could simultaneously detect both successive and computable irregularities. It uses FT-Tree to analyse logs and has developed template2vec, a new word representation method that uses synonyms and antonyms to effectively discover anomalies. When only the log template index is evaluated in [2], and the semantic log connection cannot be provided, this solution tackles the issue of losing key log information. Xiaojuan Wang et al. [3] used NetEngine40E to collect router logs and analyse behaviour type, attributes, and rank.

The projected model is an LSTM neural network that analyses the amount of logs over time to forecast log spikes. The Aspect syntax forest is also utilised for attribute data semantic analysis. The work has been extended to identify logs that are the cause of log spikes based on attribute information and value. "Robust Log," one of the most current log anomaly detection approaches, was suggested by Xu Zhang et al. [8]. They built a BiLSTM classification model based on the vector demonstration of each log event and the semantic information included in the log's semantic vector. FastText [36] word vectorization and TF-IDF-based aggregation are utilised to generate log event vectors. As demonstrated by the development of a synthetic HDFS log dataset, the robust-log appears to operate well in unstable log events. The issues presented by academics, as well as the Deep Learning models, datasets, and approaches used in various log analysis research projects, are summarized in Table 1.

Table 1 “Summary of Challenges Addressed and methods Used By Various Authors on Log Anomaly Detection”

Year	Citation	Challenges Addressed	DL Model	Data Set	NLP /Other Method	Pre-Processing / Parsing
2019	Xiaojuan Wang [3]	Time period extracted Semantics were also expressed by anomalies and reasons of Log surge.	LSTM	Netengine 40E Router Log	Directed Graph	Parsed on Behavior type
2019	Xu Zhang [8]	Relevant knowledge of log sequences was aided by log data instability.	Bi-LSTM with Attention	HDFS, Other security system of Microsoft	FastText	Drain
2019	WeibinMeng[4]	When only logtemplates are used, it is possible to discover both sequential and quantitative anomalies at the same time.	LSTM	BGL, HDFS	Template2Vec	FT-Tree
2019	Amir Farzad [6]	In log message classification and anomaly detection, LSTM and Bi-LSTM models with autoencoders are utilised.	Auto-LSTM, Auto-BLSTM,Auto-GRU	BGL, IMDB, Open stack, Thunderbird	Word Frequency	-
2018	Siyang Lu [5]	The performance of CNN with LSTM and Multilayer Perceptron for log anomaly detection was compared (MLP)	CNN based model	HDFS	LogKey2Vec	Logs-Key Sequences and session key
2018	Andy Brown[9]	Concentration has an effect on sequence modelling.	LSTM with 5 attention mechanism	LANL, cybersecurity dataset	-	Language Modeling and Tokenization
2018	Mengying Wang [1]	For log anomaly detection, NLP approaches such as word2vec and TFIDF are used.	LSTM	Thunderbird	Word2Vec , TF-IDF	Data Cleaning of Logs
2017	Min Du [2]	Workflows are used to detect and analyze anomalies based on "Log Key" and "Parameter Value."	LSTM	HDFS, Openstack	Log Key, Parameter value and Workflow	Spell

III. Specific Attack Detection/Prevention

In our research, we found that some of the studies reviewed focus on protecting web servers from specific types of attacks, mainly DDoS and Injection Attacks.

A list of the most studied types of attacks is presented below; in addition, Table 5 details the specific attacks, the number of studies dealing with each particular attack, and a list of the relevant citations.

Table 2. Detail of attacks.

Attack	Number of Studies
DDos	11
Injection	10
Botnets	2
Defacement	2
Other Attack	62

3.1 DDoS Attacks

Denial of Service (DoS) attacks are a form of attack that seeks to make a network resource unavailable by overloading the resource or machine with an overwhelming number of packets, crashing or severely slowing resource performance. Distributed Denial of Service (DDoS) is a large-scale, internet-distributed DoS attack.

In a first phase, the attacker identifies and exploits vulnerabilities in one or more networks to install malware programmes on multiple computers for remote control. At a later stage, these compromised computers are exploited for the mass sending of attack packets to the target(s) usually outside the original computer network. These attacks occur without compromised hosts knowledge.

Thang and Nguyen[32] proposed a framework for detecting DDoS attacks; this framework was based on using an online scanning process to detect certain DDoS attack traits and build a dynamic blacklist. Tripathi and Hubballi[33] proposed using chi-square tests to detect slow denial of service attacks against HTTP/2 protocol. In [34], Najafabadi et al. proposed a method for detecting application layer DDoS attacks that worked to extract instances of user behaviours requesting resources from HTTP web server logs and using Principal Component Analysis (PCA) to detect anomalous behaviour. Zolotukhin and Kokkonen[35] focused on detecting application-layer DoS attacks using encrypted protocols by applying an anomaly-detection-based approach to statistics extracted from headers of network packets using the stacked auto-encoder algorithm. Shirani, Azgomi, and Alrabaee proposed detecting DDoS attacks on Web Services using time series and ARIMA model. In training and testing phases, Tripathi, Hubballi and Singh[37] used Hellinger's distance between two probability distributions to detect Slow HTTP DoS attacks. Wang et al. proposed a sketch-based anomaly detection scheme for DDoS applications. The scheme uses sketch divergence in two consecutive detection cycles to detect an anomaly, designing a Hellinger Distance variant to measure the divergence to mitigate the impact of network dynamics. Wang et al.[39] proposed multi-feature entropy prediction model information to prevent flooding App-DDoS attacks; a second-order Markov detection model was proposed for asymmetric attacks.

Xie and Tang [40] proposed a web user behaviour model to detect Hidden Markov Model-based DDoS attacks. Markov states represent users' click-behaviour, while different states represent hyperlinks between pages.

Lin et al.[41] proposed a new statistical model for detecting DDoS attacks called Rhythm Matrix (RM), based on the packet size and intervals of consecutive HTTP request packets, in a flow indicating the behaviour of users when opening and browsing web pages.

RM characterised distribution of user access trajectory fragments, including order of visiting pages and time spent on each page. Change rate abnormality in the RM was used to detect DDoS attacks and further identify malicious hosts based on their RM drop points.

3.2 Injection Attacks

Injection flaws allow an application to send malicious code to another system. Examples of these attacks are calling the operating system via commands, using external applications via shell commands, and calling backend databases via SQL (i.e. SQL injection).

SQL Injection (SQLI) is a typical online application attack. An attacker may have direct access to the applicable database by exploiting poor input validation [121].

To identify SQLI and Cross Site Scripting (XSS) assaults, Kozik, Choras and Holubowicz [43] used non-supervised token extraction from HTTP requests as well as evolutionary token alignment. Wang et al.[44] introduced FCER Mining as a novel method for mining frequently closed episode rules in massive data on Spark to quickly locate valid rules.

They ran several SQLMAP map tool tests to see if the proposed method could withstand SQLI assaults. Yuan et.al. [45] proposed three-step strategy to detect and prevent SQLI attacks: To begin with, an ensemble clustering model differentiates abnormal samples from normal samples. In the second step, semantic anomaly presentations are obtained using word2vec technique.

Finally, another multi-clustering method categorises anomalies. Kozik, Choras, and Holubowicz [49] suggested a modified Linear Discriminant Analysis (LDA) methodology for detecting SQL injection attacks, which included reducing dimensionality using Singular Value Decomposition (SVD) and adapting Simulated Annealing for LDA vector projection.

3.3 Botnets Attacks

A bot is a compromised computer that can execute its master's commands, and bots are networked into a botnet with topology chosen by their master.

Botnet differences than other types of attacks is the existence of Command and Control (C&C) that work in giving bot-to-bot orders. Bots always hide in search of an unattended target, when bot finds the target they report to the botmaster[123]. Yu, Guo, and Stojmenovic [53] created a four-parameter semi-Markov model for browsing behaviour. Based on this model, they found it impossible to detect imitating statistical attacks if the number of active bots of the attacking botnet is sufficiently large (though it is hard for botnet owners to satisfy the condition to carry out a mimicking attack most of the time). They concluded that mimicking attacks could discriminate with second-order statistical metrics from genuine flash crowds, defining a new correntropy metric. Sakib and Huang[54] proposed detecting HTTP-based C&C traffic using statistical features based on client-generated HTTP request packets, and response packets generated by DNS server.

3.4 Defacement

A bot is a compromised computer that can execute its master's commands, and bots are joined to form a botnet[122]. Botnets are distinguished from other types of assaults by the presence of Command and Control (C&C), which provides botmaster-to-bot orders. When hunting for an unattended victim, bots are always hidden, and when they find one they report to the botmaster[123]. Yu, Guo, and Stojmenovic[53] created a semi-Markov four-parameter model to describe browsing behaviour.

They observed that if the attacker botnet has enough active bots, imitation attacks cannot be detected using data (though it is hard for botnet owners to satisfy the condition to carry out a mimicking attack most of the time).

They concluded that statistical metrics of second order can be used to separate imitation attacks from genuine flash crowds, leading to a new correntropy metric being developed. Sakib and Huang[54] recommended using statistical features based on client-generated HTTP request packets and DNS server-generated response packets to identify HTTP-based C&C activity. Anomalies were found using Chebyshev's Inequality, One-Class Support Vector Machines (OCSVM) and Nearest Neighbor Local Outlier Factor.

3.5. Other Attacks

This group includes all those studies in which the type of attack being studied is not clearly specified either because it uses non-publicly accessible datasets and does not provide information about the type of attack it attempts to detect, or because it does not attempt to detect a specific type of attack but any web request that is considered anomalous, etc.

IV. CONCLUSION

In this work, a systematic review of the published studies on the detection of web attacks using anomaly detection techniques has been carried out. One of the major drawbacks detected in this systematic review is the unavailability of a standardized, updated and correctly labelled dataset, which allows the verification of the experimental results obtained in the different studies. A small number of studies in which dimensionality reduction techniques are applied have also been detected. Dimensionality reduction allows the analysis of a larger amount of data in a shorter period of time, simplifying the complexity of sample spaces with many dimensions while preserving their information.

A combination of two or more clustering and/or classification algorithms is common. A reduced use of classic metrics is detected in works related to vulnerability detection, such as F-Score, accuracy and ROC/AUC metrics such as FPR, DR/Accuracy and TPR are widely used. Although these last metrics may be valid, the authors believe that more research efforts should be made in this area, in order to establish a concrete methodology that facilitates the choice of particular metrics depending on the type of study being conducted.

In the review of the studies carried out, it was found that most of them do not clearly specify the type of attack they are trying to prevent, although there are a small number that investigate DDoS, injection, botnets and defacement attacks. Further research efforts may be needed to generate studies that investigate the prevention and detection of other types of attacks.

In general, high statistical performance is observed in the various papers that incorporate deep learning techniques; however, in those that report on the datasets used, it is discovered that the results are highly dependent on the datasets, with the accuracy percentage decreasing as the dataset becomes more recent.

Therefore, in addition to encouraging the generation and use of public datasets to replicate and validate experiments as mentioned above, it should be further analysed whether deep learning really improves intrusion detection systems. Applying deep learning for log analysis is a rapidly growing approach to gaining knowledge from unstructured text log messages.

This study is an extension of the log-anomaly consolidation work using Deep Learning. In this research, we examined several deep learning techniques for log detection. We also summarised several NLP extraction approaches used to gather semantic and context log message information.

In recent years, automated log analysis and anomaly detection technologies have been intensively investigated to reduce manual work. However, developers are still not aware of state-of-the-art anomaly detection methods, and sometimes, due to the lack of a complete evaluation and comparison of existing approaches, they need to re-design a new anomaly detection method.

References:

[1] Liao, H.J.; Richard Lin, C.H.; Lin, Y.C.; Tung, K.Y. Intrusion detection system: A comprehensive review. J. Netw. Comput. Appl. 2013, 36, 16–24.

[2] Jyothsna, V. A Review of Anomaly based Intrusion Detection Systems. Int. J. Comput. Appl. 2011, 28, 26–35.

[3] Kakavand, M.; Mustapha, N.; Mustapha, A.; Abdullah, M.T.; Riahi, H. A Survey of Anomaly Detection Using Data Mining Methods for Hypertext Transfer Protocol Web Services. JCS 2015, 11, 89–97.

[4] Samrin, R.; Vasumathi, D. Review on anomaly based network intrusion detection system. In Proceedings of the 2017 International Conference on Electrical, Electronics, Communication, Computer, and Optimization Techniques (ICEECCOT), Mysuru, India, 15–16 December 2017; pp. 141–147.

[5] Kitchenham, B.; Charters, S. Guidelines for Performing Systematic Literature Reviews in Software Engineering Version 2.3; Technical Report; Keele University: Keele, UK; University of Durham: Durham, UK, 2007.

[6] Brereton, P.; Kitchenham, B.A.; Budgen, D.; Turner, M.; Khalil, M. Lessons from applying the systematic literature review process within the software engineering domain. J. Syst. Softw. 2007, 80, 571–583.

[7] Budgen, D.; Brereton, P. Performing Systematic Literature Reviews in Software Engineering. In Proceedings of the 28th International Conference on Software Engineering, Shanghai, China, 20–28 December 2006; Association for Computing Machinery: New York, NY, USA; pp. 1051–1052.

[8] Kitchenham, B.; Pearl Brereton, O.; Budgen, D.; Turner, M.; Bailey, J.; Linkman, S. Systematic literature reviews in software engineering—A systematic literature review; Inf. Softw. Technol. 2009, 51, 7–15.

[9] Kitchenham, B.; Brereton, P. A Systematic Review of Systematic Review Process Research in Software Engineering. Manuscr. Publ. Inf. Softw. Technol. 2013, 55, 2049–2075.

[10] Patel, A.; Taghavi, M.; Bakhtiyari, K.; Celestino Júnior, J. An intrusion detection and prevention system in cloud computing: A systematic review. J. Netw. Comput. Appl. 2013, 36, 25–41.

[11] Raghav, I.; Chhikara, S.; Hasteer, N. Article: Intrusion Detection and Prevention in Cloud Environment: A Systematic Review. Int. J. Comput. Appl. 2013, 68, 7–11.

[12] Patcha, A.; Park, J.M. An overview of anomaly detection techniques: Existing solutions and latest technological trends. Comput. Netw. 2007, 51, 3448–3470.

[13] Chandola, V.; Banerjee, A.; Kumar, V. Anomaly Detection: A Survey. ACM Comput. Surv. 2009, 41.

[14] Jose, S.; Malathi, D.; Reddy, B.; Jayaseeli, D. A Survey on Anomaly Based Host Intrusion Detection System. J. Phys. Conf. Ser. 2018.

[15] Fernandes, G.; Rodrigues, J.J.P.C.; Carvalho, L.F.; Al-Muhtadi, J.F.; Proença, M.L. A comprehensive survey on network anomaly detection. Telecommun. Syst. 2019, 70, 447–489.

[16] Kwon, D.; Kim, H.; Kim, J.; Suh, S.C.; Kim, I.; Kim, K.J. A survey of deep learning-based network anomaly detection. Clust. Comput. 2019, 22, 949–961.

[17] Tavallaee, M.; Bagheri, E.; Lu, W.; Ghorbani, A.A. A Detailed Analysis of the KDD CUP 99 Data Set. In Proceedings of the Second IEEE International Conference on Computational Intelligence for Security and Defense Applications, Ottawa, ON, Canada, 8–10 July 2009; IEEE Press: Piscataway, NJ, USA, 2009; pp. 53–58.

[18] McHugh, J. Testing Intrusion Detection Systems: A Critique of the 1998 and 1999 DARPA Intrusion Detection System Evaluations as Performed by Lincoln Laboratory. ACM Trans. Inf. Syst. Secur. 2000, 3, 262–294.

[19] Mahoney, M.V.; Chan, P.K. An Analysis of the 1999 DARPA/Lincoln Laboratory Evaluation Data for Network Anomaly Detection BT—Recent Advances in Intrusion Detection. In Recent Advances in Intrusion Detection; Vigna, G., Kruegel, C., Jonsson, E., Eds.; Springer: Berlin/Heidelberg, Germany, 2003; pp. 220–237.

[20] Brugger, T. KDD Cup '99 dataset (Network Intrusion) considered harmful. KDnuggets News 2007, 7, 15.

[21] Ieracitano, C.; Adeel, A.; Gogate, M.; Dashtipour, K.; Morabito, F.C.; Larijani, H.; Raza, A.; Hussain, A. Statistical Analysis Driven Optimized Deep Learning System for Intrusion Detection BT. In Advances in Brain Inspired Cognitive Systems; Ren, J., Hussain, A., Zheng, J., Liu, C.L., Luo, B., Zhao, H., Zhao, X., Eds.; Springer International Publishing: Cham, Switzerland, 2018; pp. 759–769.

[22] Ieracitano, C.; Adeel, A.; Morabito, F.C.; Hussain, A. A novel statistical analysis and autoencoder driven intelligent intrusion detection approach. Neurocomputing 2020, 387, 51–62.

[23] Khraisat, A.; Gondal, I.; Vamplew, P.; Kamruzzaman, J. Survey of intrusion detection systems: Techniques, datasets and challenges. Cybersecurity 2019, 2, 20.

[24] Ahmed, M.; Naser Mahmood, A.; Hu, J. A survey of network anomaly detection techniques. J. Netw. Comput. Appl. 2016, 60, 19–31.

[25] Kotu, V.; Deshpande, B. Chapter 13 Anomaly Detection. In Data Science, 2nd ed.; Kotu, V., Deshpande, B., Eds.; Morgan Kaufmann: Burlington, MA, USA, 2019; pp. 447–465.

[26] Hodge, V.J.; Austin, J. A Survey of Outlier Detection Methodologies. Artif. Intell. Rev. 2004, 22, 85–126.

[27] Kaelbling, L.P.; Littman, M.L.; Moore, A.W. Reinforcement learning: A survey. J. Artif. Intell. Res. 1996, 4, 237–285.

[28] Guyon, I.; Elisseeff, A. An Introduction to Variable and Feature Selection. J. Mach. Learn. Res. 2003, 3, 1157–1182.

[29] Pudil, P.; Novoviˇcová, J. Novel Methods for Feature Subset Selection with Respect to Problem Knowledge BT—Feature Extraction, Construction and Selection: A Data Mining Perspective. In Feature Extraction, Construction and Selection. The Springer International Series in Engineering and Computer Science; Liu, H., Motoda, H., Eds.; Springer: Boston, MA, USA, 1998; Volume 453; pp. 101–116._7.

[30] Hu, H.; Zahorian, S.A. Dimensionality reduction methods for HMM phonetic recognition. In Proceedings of the 2010 IEEE International Conference on Acoustics, Speech and Signal Processing, Dallas, TX, USA, 14–19 March 2010; pp. 4854–4857.