JOURNAL OF ALGEBRAIC STATISTICS
Volume 13, No. 1, 2022, p. 129-148
https://publishoa.com
ISSN: 1309-3452

Anomaly Detection for Web Log Data Analysis: A Review

*Meena Siwach & **Dr. Suman Mann
**Research Scholar, USICT, GGSIPU, Dawraka , New Delhi*
**Assistant Professor, Maharaja Surajmal Institute of Technology, Janakpuri, Delhi*
*** Associate Professor, Maharaja Surajmal Institute of Technology, Janakpuri, Delhi*
**meenusiwach@gmail.com*
** *sumanmann@msit.in*

Received 2022 March 15; **Revised** 2022 April 20; **Accepted** 2022 May 10.

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; in particular, we utilised Kitchenham's standard approach for conducting a organized analysis of literature in the computer science area. There are 8041 peer-reviewed publications published in major journals. This technique is used to 88 articles. This page outlines the processes taken to perform this systematic review, as well as the findings and conclusions made. The majority of logs are utilised for anonymous detection and recording system runtime data. Developers (or operators) used to manually examine logs by looking for keywords and matching rules. However, as the size and complexity of contemporary systems grows, the number of logs grows exponentially, making manual testing unfeasible. Many techniques of anomaly identification for automated log analysis have been suggested to minimise 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, Auto encoder, CNN, Deep Learning, LSTM, Log Parsing

1. Introduction

Due to various their high value, Web servers are gradually becoming targets for assaults as the information technology sector advances. SQL injection and cross-site scripting (XSS) threats have been increasingly common in recent years, which is why Web security has received more attention from academic and industry communities. Anomaly is a term used in internet security research. The analysis of log data is used in web detection. Log files, as crucial recording data, may reveal extensive information at the time of system operation and may be used to trace the majority of assaults. However, log systems create a lot of data, and critical information might be lost in the shuffle.

Furthermore, due to the ever-changing nature of assaults and hacking techniques, gathering anomaly data has become increasingly complex, leading to the current problem that manual log file analysis is inadequate to meet log testing standards. In addition, conventional intrusion detection techniques involve operators or programmers to remove the attack features manually, and detect common attack patterns depend on searching of keyword and rule matching [1]. In another words, the conventional approach cannot detect unidentified attacks and leads to fail.

A variety of anomaly detection techniques are being suggested to solve the limitations of conventional methods in order to overcome the shortcomings of previous years. Many machine learning methods are being utilised in the identification of log-based anomalies as a result of advances in machine learning [2]. Anomaly detection techniques are typically split into two groups based on the kind of data and the use of machine learning technology: supervised detection [3] and

unsupervised detection [4, 5].Normal training data, properly described in both happy and bad circumstances, is required for the supervised approach. Unsupervised techniques, on the other hand, do not need labels at all. Their research is based on the fact that a traumatic experience may sometimes act like a far-fetched advertisement.

In this paper, machine learning based system of anomaly detection is proposed for Weblog file. To reduce the above-mentioned disadvantages of traditional method, this system uses an algorithm of Machine learning on two levels. The decision tree classifier is used to choose standard log files, and then Markov's hidden model is used to build a standard data model set (hereafter HMM).This is an example of a model for detecting anomalies. It generates automated knowledge and training based on a large amount of data, raising the stakes in the online security debate to new heights. With an accuracy of 93.54 percent and an error rate of 4.09 percent, an examination of 4,690,000 messages from real-time industrial and real-time import samples demonstrates the efficacy of our anomaly detection technology.

However, to our knowledge, no open-source log-based anomaly detection technologies are presently available. Also, there is a deficiency of comparison between the various approaches to detecting anomalies. Engineers have a tough time determining the optimum approach to their practical challenges. They must attempt each potential method with their own implementations in order to compare them. Because there are no test oracles to guarantee that the underlying machine learning algorithms are implemented correctly, reproducing the techniques frequently necessitates a significant amount of work. To fill this need, we present a thorough analysis and evaluation of log-based anomaly identification in this study, as well as the availability of an open-source toolkit 1 for anomaly identification. Our objective is to provide a complete overview of current anomaly detection research in log analysis, rather than to enhance any one approach. The key processes in the log analytical process for anomaly detection are log collection, log division, feature elimination, and anomaly detection. We recently published a study and assessment of automatic log parsing techniques [24], which included the public release of four open-source log parsers. A well-thought-out training data set with exact definitions of normal and abnormal cases is required for supervised anomaly detection. The model is then learned using classification approaches to maximise the discrimination between normal and abnormal cases. Unsupervised approaches, on the other hand, do not require labels. They are based on the notion that an aberrant instance is generally presented as a faraway outlier point from other examples. As a result, techniques like clustering and unsupervised learning may be used. Six common anomaly detection strategies from the literature were explored and implemented, including three supervised" (Logistic Regression [12], Decision Tree [15], and SVM [26]) methods (i.e., Log Clustering [27], PCA [47], and Invariant Mining [28])". After that, the methods were put to the test on two publicly available log datasets, which included 15,923,592 log messages and 365,298 anomalous occurrences. (See https://github.com/cuhk-cse/loglizer for further information). The evaluation's findings are offered in terms of accuracy (percentage of accurately reported anomalies), recall (percentage of genuine abnormalities detected), and efficacy (percentage of true abnormalities found) (in terms of the running times over different log sizes). Despite the limited data, we believe that these findings, along with the accompanying conclusions, can be used as recommendations for implementing these methodologies and as guidelines for further development. In conclusion, the following contributions were made by this paper:

- A method for detecting weblog files has been presented as an anomaly detection system.
- After comparing many machine learning algorithms and discovering that this anomaly detection system has a small number of facts to discover the truth without giving a solid precision, the system uses a two-level machine learning algorithm, a decision tree algorithm, and HMM to detect undesirable data and anonymous attacks.
- Weblogs are categorized into real-world industrial scenarios with several instances of actual assaults, implying that the data setup is widespread and successful.

2. 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.

JOURNAL OF ALGEBRAIC STATISTICS
Volume 13, No. 1, 2022, p. 129-148
https://publishoa.com
ISSN: 1309-3452

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.

2.1 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.

2.2 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.

2.3 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.

JOURNAL OF ALGEBRAIC STATISTICS
Volume 13, No. 1, 2022, p. 129-148
https://publishoa.com
ISSN: 1309-3452

3. Data Set Used

Log Datasets: Firms rarely publish production logs due of privacy concerns. Our research yielded two log datasets, HDFS data[47] and BGL data[36], appropriate for evaluating existing anomaly detection algorithms. They contain a total of 15,923,592 log messages and 365,298 anomaly samples. We consider these designations (anomaly or not) to be ground truth.

Additional data about statistical data sets can be found in Table I. There are 11,175,629 log messages on the Amazon EC2 platform[47]. Each block operation in HDFS logs has a unique allocation, writing, replication, and deletion block ID. As a result, session windows, as introduced in III-B, can capture log operations more naturally, as each distinctive block ID can be utilised to break logs into a number of log sequences. We then create 575,061 event count vectors by extracting vectors from these log sequences. There are 16,838 samples that are considered to be random. The LLNL Blue Gene/supercomputer L's system gathered 4,747,963 log messages in BGL data[36]. BGL logs, unlike HDFS data, do not have a unique identification for each task. To segment logs, we must first create log sequences using fixed windows or external doors, and then take out the relevant event count vectors. However, the amount of windows is determined by the window's size (and step size). BGL data failures account for 348,460 log messages, and every log sequence that contains any failure records is considered as an anomaly.

Table 2: Word types that correspond to data of various sizes in various datasets

Log Sets	Word types corresponding to different sizes of data		
	1M	100M	1G
Liberty	9112	436777	2368053
HDFS	4538	211080	2459515
Spirit	8631	454437	3055695
Thunderbird	13444	264629	1165458
Zookeeper	5590	53094	53094
BGL	10904	845629	4592728

Table 3 Summary of Dataset

System	#Time span	#Data size	#Log message	#Anomalies
HDFS	38.7 hours	1.55 G	11,175,629	16,838
BGL	7 months	708 M	4,747,963	348,460

The sensitivity of the data explains why there isn't any publicly available data: Inspection of network traffic might depict highly sensitive information about a company. Researchers are obliged to create their own datasets due to a shortage of publicly available data, often without access to suitably sized networks: behaviour observed in a small laboratory network cannot be extrapolated to a larger network[167]. The following are the public datasets that were used in various studies for the planned experiments:

- DARPA: DARPA was founded in 1998 by the Massachusetts Institute of Technology's Lincoln Laboratory. A new version of the dataset (containing fresh attacks and a Windows NT target) was generated in 1999 due to fast changing technology [168]. DARPA98 and DARPA99 are raw TCP dump files that allow 244 marked occurrences of 58 different attacks to be tested on four operating systems (SunOS, Solaris, Linux, and Windows NT). Several scholarly articles [18,19] have slammed these datasets, mostly because they were created using false data: specialised software was employed to synthesise user behaviour and typical network traffic to create a small, insulated network that looked like it belonged to the Air Force. Rendering to McHugh [18], the dataset has traffic data collecting issues, such as a shortage of statistical proof of resemblance to regular Air Force network traffic (particularly in terms of false alert rate), taxonomy and distribution attacks, and evaluation standards.
- KDD Cup 99: A modified version of the DARPA dataset that includes 41 features suitable for machine learning classification methods. This data set is available in three distinct forms: a full training set, a 10% training set, and a test

JOURNAL OF ALGEBRAIC STATISTICS
Volume 13, No. 1, 2022, p. 129-148
https://publishoa.com
ISSN: 1309-3452

data set. In increasingly widespread circumstances, record duplication of training and test sets may result in skewed outcomes [17].

- NSL-KDD: developed in 2009 by the University of New Brunswick's Information Security Center of Excellence (ISCX) to address the issue of duplicate records discovered in KDD Cup 99 [17]. Due to the duplicacy of records, learning algorithms may produce biased findings, as well as a deficiency of knowledge of scarce records. The record set of 4,900,000 and 2,000,000 was reduced to 125,973 and 22,544 respectively after applying the cleanup methods in the new NSL-KDD training and test data sets.
  With the "IXIA Perfect Storm programmed, the Australian Cyber Security Center (ACCS) created a combination of real modern normal activities and contemporary synthetic attack behaviours. Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode, and Worms are among the nine types of assaults that can be launched There are 175,341 and 82,332 recordings in training sets. The simulation began on January 22, 2015 at 16 a.m. and finished on February 17, 2015 at 15 a.m..
- Kyoto 2006: The KDD Cup '99 dataset and the NSL-KDD dataset were created using a virtual network simulation and do not represent real computer network data flow. Kyoto 2006+ is based on three years of traffic data from November 2006 to August 2009. This information is gathered via honeypots, darknet sensors, site crawlers, and email servers .
- ISCX: Shiravi et al. devised a method for producing datasets for analysing and assessing intrusion detection systems, which primarily relies on anomaly detection techniques. Researchers should create datasets from a collection of summaries that may be combined to create a range of datasets. The ISCX (Information Security Center of Excellence) dataset is the product of this effort". This dataset comprises one week's worth of simulated traffic, with each record including 11 distinct characteristics. The data is labelled to distinguish between legitimate and malicious network activities.
- CSIC-2010: The CSIC 2010 dataset covers traffic produced by the Spanish National Research Council's e-commerce web application (CSIC). Users can use a shopping cart to purchase things and register personal information on this web application. The dataset was constructed automatically, containing 36,000 regular queries and over 25,000 aberrant requests . Requests are labelled as normal or anomalous. Table 4 lists the public datasets used in the studies examined, whereas Figure 1 shows the fraction of data sets utilized in the studies examined.

Table 4. Public datasets used

Dataset	No. of Studies	References
NSL-KDD	2	[75,80]
KDD-Cup 99	8	[58,78,80,81,91,97,106,116]
ISCX	1	[116]
ECML/PKDD 2007	1	[114]
UNSW-NB15	2	[75,115]
Kyoto 2006	2	[80,116]
DARPA	4	[77,82,100,102]
CSIC 2010	13	[43,59,64,65,72,74,80,85–87,93,114,117]

JOURNAL OF ALGEBRAIC STATISTICS
Volume 13, No. 1, 2022, p. 129-148
https://publishoa.com
ISSN: 1309-3452

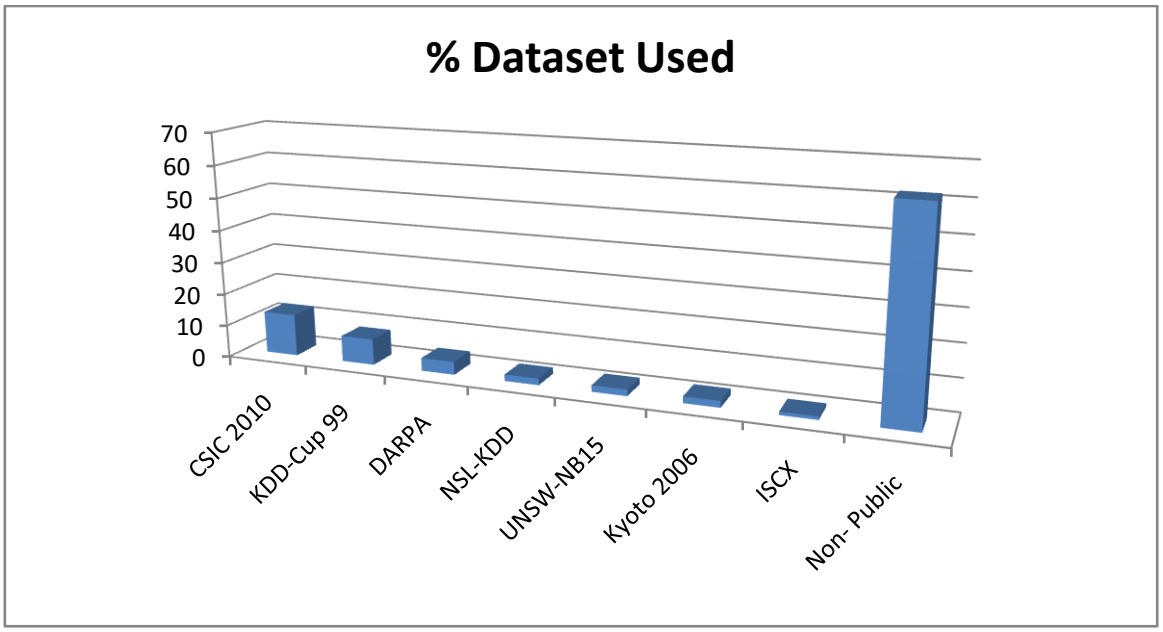

Figure 1. % of datasets used.

4. Specific Attack Detection/Prevention

We discovered during our research that several of the studies we looked at focused on protecting web servers from certain sorts of assaults, such as DDoS and Injection Attacks. Table 5 shows the specific attacks, the number of research dealing with each attack, and a list of pertinent citations.

Table 5. Detail of attacks.

Attack	Number of Studies	Citation
DDos	11	[32-42]
Injection	10	[43-52]
Botnets	2	[53,54]
Defacement	2	[55,56]
Other Attack	62	[57-118]

4.1 Denial of Service (DOS) Attacks

DoS attacks attempt to make a network resource inaccessible by flooding the resource or computer with an excessive amount of packets, causing the resource to crash or significantly slow down. DDoS (Distributed Denial of Service) is a large-scale, internet-wide denial-of-service attack. In the first phase, the attacker detects and exploits vulnerabilities in one or more networks in order to remotely control multiple computers by installing malware programmes on multiple systems. "These compromised systems are then used to transmit a large number of attack packets to the target (s), which is usually outside the original computer network. These attacks are carried out without the awareness of the hosts [119]. A system for detecting DDoS assaults was proposed by Thang and Nguyen [32]. This technology relied on an online scanning procedure to identify specific DDoS attack characteristics and create a dynamic blacklist. HTTP/2 protocol delayed denial of service attacks can be detected using chi-square tests [33]. It extracts instances of user behaviour requesting resources from HTTP web server logs and detects odd behaviour using Principal Component Analysis (PCA), as proposed by Najafabadi et al. in [34]. (PCA). Zolotukhin and Kokkonen [35] focused on detecting application-layer DoS attacks employing encrypted protocols using an anomaly-detection-based method. Time series and the ARIMA model were proposed by Shirani, Azgomi, and Alrabaee [36]. They used Hellinger's distance between two probability distributions to detect slow HTTP DoS attacks during training and testing. For DDoS attacks, Wang et al. [8] suggested a sketch-based anomaly detection technique. To limit the impact of network dynamics, the approach leverages sketch

JOURNAL OF ALGEBRAIC STATISTICS
Volume 13, No. 1, 2022, p. 129-148
https://publishoa.com
ISSN: 1309-3452

divergence in two successive detection cycles to detect an anomaly, with a Hellinger Distance version to measure the divergence. Wang et al. [39] created a second-order Markov detection approach to detect asymmetric App-DDoS attacks. HMM-based DDoS assaults can be detected using Xie and Tang's [39]. Rather than Markov states, connections between sites are represented by separate states. Based on packet size and intervals of consecutive HTTP request packets in a flow that reflects how consumers view and explore online sites." Lin et al. [41] developed the Rhythm Matrix (RM). The distribution of user access trajectory pieces is described by RM, which includes the order in which pages are visited and the amount of time spent on each page. DDoS attacks were detected and malicious hosts were identified based on their RM drop points using change rate irregularities in the RM.

4.2 Injection Attacks

This allows a programme to communicate harmful code. Using shell commands to access external programmes or backend databases are examples of these attacks (i.e. SQL injection). SQL Injection (SQLI) is a common online attack. Poor input validation may allow an attacker to gain direct database access (121). Kozik, Choras, and Holubowicz [3] employed non-supervised token extraction and evolutionary token alignment to detect SQLI and XSS attacks. Wang et al. [44] proposed FCER Mining as a new method for fast locating valid rules in vast data on Spark. It was tested using the SQLMAP map tool against SQLI attacks. Yuan et al. [15] proposed a 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 the 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.

4.3 Botnet Attacks

A bot is a compromised computer that can execute its master's commands, and bots are connected in a botnet according to the master's topology. Due to the presence of Command and Control (C), which sends bot-to-bot commands, botnets are unique sorts of attacks. Bots always hide in the hope of finding an unattended victim to report to the bot-master [23]. A 4 parameter semi-Markov model for browsing behaviour was developed by Yu, Guo, and Stojmenovic [53]. Statistics assaults are impossible to detect if the attacker botnet has a high enough number of active bots (though it is hard for botnet owners to successfully carry out a mimicking attack most of the time).

4.4 Defacement

A bot is a hijacked computer that can execute commands from its master, and botnets are composed of bots[122][123]. Botnets are distinguished from other forms of attacks by the presence of Command and Control (C & C), which communicates bot-master-to-bot orders to the bots. Bots are always hidden when seeking for an unattended victim, and when they discover one, they report it to the bot-master [125]. They constructed a semi-Markov four-parameter model to represent browsing behaviour. Unless the attacker's botnet has a large number of active bots, data cannot detect imitation attacks. They came to the conclusion that statistical metrics of the second order can be utilised to distinguish genuine flash crowds from imitation attacks, prompting the development of a new correlation metric. For detecting HTTP-based C & C, Sakib and Huang [54] suggested using statistical features derived from HTTP request and response packets. Anomalies were found using Chebyshev's Inequality, OCSVM, and the Nearest Neighbor Local Outlier Factor.

4.5 Additional Attacks

The use of non-publicly accessible datasets and the failure to provide information about the type of attack it attempts to discover, or the failure to attempt to detect a specific type of attack but rather any strange web request, are all examples of what is considered unethical research.

5. Framework of Methodology

Describe the overall structure for detecting anomalies in log files. Anomaly detection is primarily accomplished through four steps: log collecting, log processing, feature extraction, and anomaly detection.

JOURNAL OF ALGEBRAIC STATISTICS
Volume 13, No. 1, 2022, p. 129-148
https://publishoa.com
ISSN: 1309-3452

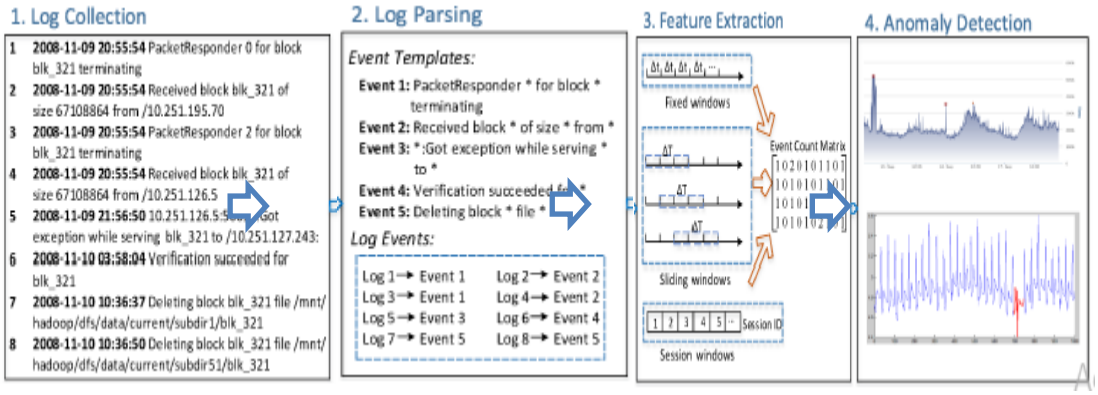

Figure 2: Framework of anomaly detection

5.1 Log collection

Logs are generated by large-scale systems in order to record system status and runtime data. Each log comprises a date and a description. The logs are saved for subsequent use because they contain vital information (e.g., anomaly detection). Figure 1 shows eight log lines from Amazon EC2 HDFS logs [47], with some information deleted for clarity. Log parsing organises raw logs by extracting templates. Each log message can be converted into an event template (constant component) that has certain parameters (variable part). The event template "Received size block * from *" is used to process the fourth log message (Log 4) as shown in Figure 1. Extraction of characteristics: We must encode logs into numerical feature vectors, which can then be employed with machine learning models, after parsing them into individual events. We next generate an event count vector for each log sequence that shows the number of times each event occurs. A matrix or event count matrix can be produced by combining all feature vectors. We'll go over multi-phase methods including “log parsing, feature extraction, and anomaly detection” in this section. We provide a basic overview of log parsing and introduce some typical log parsers. These methodologies for producing feature vectors from parsed log events are then outlined. We focused on six exemplary anomaly detection algorithms after obtaining the feature vectors, three of which are supervised techniques and three of which are unsupervised.

5.2 Log Parsing

Logs are simple texts with elements that may vary from one instance to the next. The words "Connection from 10.10.34.12 closed" and "Connection from 10.10.34.13 closed" are considered constant parts in the logs "Connection from 10.10.34.12 closed" and "Connection from 10.10.34.13 closed," for example, because they never change, but the remaining parts are known as variable parts because they are not fixed. Although developers specify constant components in source codes, variable portions (such as port numbers and IP addresses) are sometimes dynamically generated and hence unsuitable for anomaly detection. The purpose of log parsing is to extract constants from variable items and generate a well-defined log event

Extraction feature

To extract useful data from log events for use in anomaly detection models, this stage is critical. It creates an event count matrix from the log events created during log parsing. The initial step in extracting features is to organise the log data into log sequences. Windowing a log dataset [5] is one method. As seen in Figure 1, we use fixed, sliding, and session windows. The following windows have been fixed: The log occurrence time is utilised in both fixed and sliding windows. Each fixed window's size corresponds to its duration. As seen in Figure 1, the window size is 2. Permanent windows are determined by window size. Logs displayed in the same window..

Sliding windows, Sliding windows, unlike fixed windows, have two distinct characteristics: window size and step size, and windows that open every five minutes. In general, the step size is smaller than the window size, resulting in window overlap. The window size in Figure 1 is T, and the forwarding distance is the step size.

The number of sliding windows, which are frequently larger than fixed windows, is mostly controlled by the size of the window and the size of the steps. Although logs may be replicated in several sliding windows owing to overlap, logs in the same sliding window are grouped together as a log sequence. Session windows, unlike the previous two window types, are based on IDs rather than timestamps. Identifiers are used in some log data to distinguish between different

JOURNAL OF ALGEBRAIC STATISTICS
Volume 13, No. 1, 2022, p. 129-148
https://publishoa.com
ISSN: 1309-3452

execution paths. Block allocation, writing, replication, and deletion are all captured in HDFS logs with block IDs, for example. The logs can be sorted by identification, with each session window having its unique number. An event count matrix X is created when log sequences are constructed using windowing techniques. The event count vector is created by counting the number of times each event happens in each log sequence.

5.3 Supervised detection of anomalies

A machine-learning job of developing a model is defined as supervised learning (e.g., decision tree). The need for supervised anomaly detection is labelled in the training data with normal or anomalous state labels. The model would be more accurate if the training data were labelled more accurately. We'll go over three popular supervised methods in the next sections: logistic regression, decision trees, and support vector machines (SVM).

5.4 Regression logistics

The statistical regression model is frequently used for categorization. Logistic regression calculates the probability p of all possible states to determine the instance's status (normal or anomalous). A logistic function is used to determine probability p, based on labelled training data. The logistic function could calculate all conceivable states' probability p (0 p 1) when a new instance emerges. Probability categorizes the states with the highest probability. Each log sequence generates an event count vector and a label to find anomalies. First, we developed the logistic regression model, which is a logistic function. After getting the model, we utilise the logistic function to compute the probability p of anomaly for test case X; X's label is anomalous when p is 0.5 and normal otherwise.

5.5 Decision Tree

It is a hierarchal structure where branches represent the projected state for each occurrence. Using training data, the top-down decision tree is built. The current "best" attribute, determined by the information gain attribute [23], is used to generate each tree node. For example, Figure 3's root node shows our dataset has a total of 20 occurrences. The Event 2 occurrence number is used as the "best" characteristic to divide the root node. As a result, the 20 training examples are divided into two subgroups based on this characteristic value, one with 12 examples and the other with 8. The decision tree is built using event count vectors and labels.

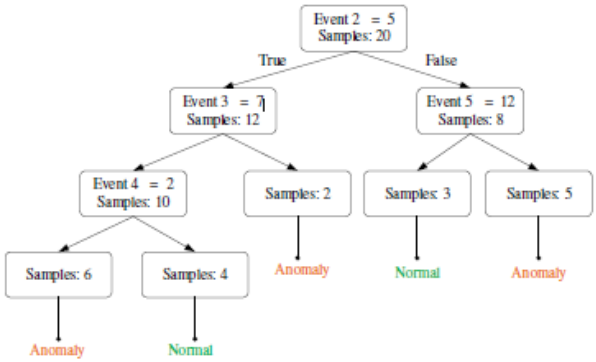

Figure 3: An example of decision tree

5.6 SVM

A supervised rating algorithm is the Support Vector Machine (SVM). A hyperplane is used in SVM to distinguish between various examples in high-dimensional space. This is an optimization problem for finding the hyperplane in various classes. SVM was used by Liang et al. in [26] to detect and compare failures. Label-like event count vectors, such as Logistic Regression and Decision Tree, are used as training cases. It is reported as an anomaly if a new instance is identified in SVM detection above the hyperplane; otherwise, it is recorded as normal. Two types of SVMs are there: linear and non-linear SVMs. We only looked at linear SVM in this study because it outperformed non-linear SVM in the majority of our tests

5.7 Unsupervised Anomaly Detection

This is a common task with unlabeled training data. With no labels, unsupervised techniques work better in real-world manufacturing. Algorithms for unsupervised learning include clustering and PCA.

JOURNAL OF ALGEBRAIC STATISTICS
Volume 13, No. 1, 2022, p. 129-148
https://publishoa.com
ISSN: 1309-3452

5.8 Cluster log

Lin et al. created Log Cluster in 1927 to identify system failures. Both the knowledge base setup and online learning phases are necessary for the Log Cluster. These two phases have two half of training examples. These are: log vectorization, log clustering, and vector-representation. Log sequences are first vectorized into event count vectors, then improved using IDF [41] and standardization procedures. Second, the Log Cluster employs agglomerative hierarchical clustering to construct two sets of vector clusters (normal and aberrant). In the end, we calculate each cluster's centroid

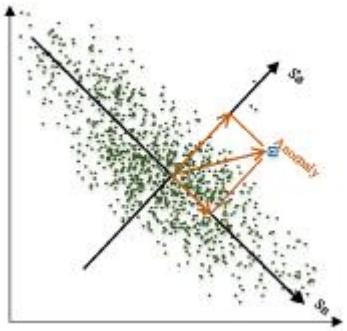

Figure 4: Anomaly detection example with PCA

It is during the online learning phase that the knowledge base clusters are fine-tuned. During the online learning phase, event count vectors are added. Calculate distances between event count vectors and representative vectors. If the event count vector is less than the threshold, the nearest cluster's representative vector will be updated. Otherwise, for a new cluster, Log Cluster uses this event count vector. The Log Cluster can be used to detect abnormalities once the knowledge base is built and the online learning process is complete. To identify its status, we calculate the distance between a new log sequence and representative vectors in the knowledge base. The log sequence is reported as an anomaly if the smallest distance exceeds a threshold. If the next cluster is normal/abnormal, the log sequence is reported otherwise.

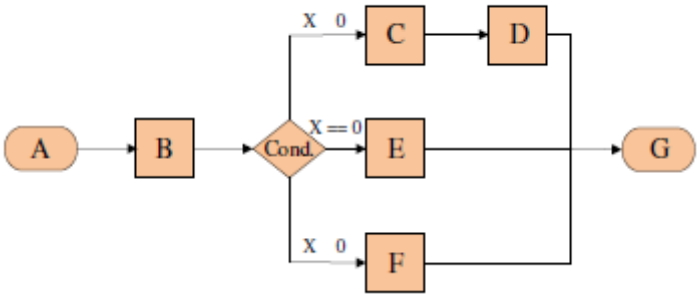

Figure 5: An example of execution flow

PCA is a "prominent statistical dimension reduction approach. Projecting high-dimension data (e.g., points) to a new coordinate system with k principal components (i.e. k dimensions) is central to PCA's concept. PCA locates the components (i.e., axes) that capture the most variation in the high-dimension data. Consequently, PCA-transformed low-dimension data may maintain primary high-dimension data qualities (for example, the similarity between two points). In Figure 3, PCA tries to convert two-dimensional points to one-dimensional. The main component is Sn, because mapping locations to Sn best describes distance. Xu et al. [47] used PCA for log-based anomaly detection. Each log sequence is vectorized as an event count vector in their detection technique." The PCA algorithm is then used to find patterns in the event count vector dimensions. PCA is used to create two subspaces: normal Sn space and anomalous Sa space. The first k primary components construct Sn, whereas the remaining (nk) construct Sn, with n being the original dimension. The

JOURNAL OF ALGEBRAIC STATISTICS
Volume 13, No. 1, 2022, p. 129-148
https://publishoa.com
ISSN: 1309-3452

projection of an event count vector y to Sa is then determined using the formula ya = (1PPT)y, with P = [v1, v2,.....vk,] as the primary k component.

If the length of ya surpasses the threshold, an anomaly will be recorded. Figure 3 shows an anomaly because Sa's projection length is excessively large. An event count vector is anomalous if

$$SPE = \|y_a\|^2 > Q_\alpha$$

SPE is the "length" and Q is the confidence level (1-). Our article has Q=0.001. In order to compute k, we tweak the PCA to incorporate data variance of 95%.

5.9 Invariants Mining

During system execution, programme invariants are linear relationships that always hold under varying workloads, even with different inputs. Invariant mining was used for the first time in log-based anomaly detection in[28]. The execution flow of a session is commonly represented by logs with the same session id. A simple programme flow is shown in Figure 5. A log message is generated at each stage of this execution flow, from A to G. Following the programme execution sequence in Figure 4, and assuming there are many instances operating in the system, the following equations are correct.

$n(A) = n(B)$
$n(B) = n(C) + n(E) + n(F)$
$n(C) = n(D)$
$n(G) = n(D) + n(E) + n(F)$

where n (*) is the number of logs belonging to the event type.

For example, n (A)=n (B)) appears to be a linear relationship between several log events showing regular system performance behaviours. Real-world system events are linear.

Opened files must be closed. Thus, "open file" and "close file" come in pairs. If the number of "open file" log events and "close file" log events in an instance are not equal, it is considered irregular. Mining invariants consists of three steps. Each row of the Invariants mining input is an event count vector. Singular value decomposition is used to estimate invariant space, which specifies the quantity of invariants to mine next.

Its invariants are then found through brute force search. A threshold is then applied to each mined invariant candidate (e.g., supported by 98 percent of the event count vectors).

To acquire independent invariants, repeat step 2. When a new log sequence arrives, we examine the invariants' obedience. There must be at least one invariant that is broken.

5.10 Methods Comparison

We explain the pros and disadvantages of different methods in this section to assist developers to better grasp the above six anomaly detection methodologies and better pick anomaly detection techniques to employ. Labels are necessary for anomaly detection in supervised algorithms. Developers may find abnormalities with reasonable explanations using the decision tree, which is more interpretable than the other two techniques (i.e., predicates in tree nodes).SVM with kernels may tackle linearly non-separable problems that are not addressed by logistic regression. However, because SVM parameters are difficult to tune (for example, the penalty parameter), establishing a model sometimes necessitates a lot of human effort. Because there are no labels, unsupervised approaches are more practical and meaningful. The concept of log clustering is based on online learning.

As a result, it's ideal for analysing massive amounts of log data. Not only can invariants mining discover anomalies with great accuracy, but it can also offer a relevant and comprehensible analysis for each anomaly found. The process of mining invariants, on the other hand, takes a long time. PCA is difficult to grasp and is sensitive to data. As a result, the accuracy of its anomaly detection varies depending on the dataset.

6. Performance Parameter

This chapter discusses the most often used metrics for evaluating the various experiments described in the evaluated literature.

Accuracy (ACC) is the clearly recognized payload ratio divided by total generated payloads.

$$ACC = \frac{TP + TN}{TP + TN + FP + FN}$$

JOURNAL OF ALGEBRAIC STATISTICS
Volume 13, No. 1, 2022, p. 129-148
https://publishoa.com
ISSN: 1309-3452

False Alert Rate (FAR) or False Positive Rate (FPR): It is the likelihood of a false alarm being raised. When the true value is negative, a positive result will be given

$$.FAR = \frac{FP}{FP+TN}$$

True Negative Rate, Specificity: It is a metric that indicates the proportion of false negatives that are genuinely identified.

$$TNR = \frac{TN}{FP + TN}$$

True Positive Rate, Recall, Sensitivity and Detection Rate: The True Positive Rate (TPR), also known as Reminder, Sensitivity, or Detection Rate (DR), is a metric that indicates the proportion of true positives that are accurately identified.

$$TPR = \frac{TP}{TP + FN}$$

Positive Predictive Value (PPV), is the percentage of accurately detected malicious payloads to total malicious payloads (PPV).

$$PPV = \frac{TP}{TP + FP}$$

False Negative Rate: It is the percentage of positive test results linked with a test, or the conditional likelihood of a negative test end result given the presence of the disease.

$$FNR = \frac{FN}{FN + TP}$$

F1-Score: The F1 score is a test accuracy metric that considers both accuracy and recall. Precision (P) and recall (R) are the weighted harmonic means of two performance measures (R) which is used to calculate the F1-Score.

$$F1 - Score = \frac{1}{\alpha.\frac{1}{p} + (1-\alpha).\frac{1}{R}}$$

Classification error: The total number of samples that were incorrectly classified was calculated is referred to as "classification error," and it is computed using the formula:

$$CE = \frac{f}{n}.100$$

Matthews Correlation Coefficient (MCC) is a measure of the quality of binary classifications (two-class). MCC is a correlation coefficient that provides a value between -1 and +1 for observed and predicted binary classifications. A coefficient of +1 denotes a flawless prediction, a coefficient of 0 denotes no improvement over random prediction, and a coefficient of -1 denotes total; disagreement between ;prediction and observation. It's the same thing as the phi-coefficient.

$$MCC = \frac{TP.TN - FP.FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

Area under Curve: The true positive rate (TPR) is plotted against the false positive rate (FPR) to show how well a classification model performs across all categorization thresholds (FPR). The AUC value is a two-dimensional region beneath the entire ROC curve that ranges from 0 to 1 (100 percent inaccurate predictions).

Table 6. Measures used in experiment evaluation

Metric	Number of Studies	Citations
MCC	1	[71]
CE	1	[124]
FNR	4	[55,56,95,100]
F-Score	12	[47,59,62,65,71,76,80,89,113–115,118]
TNR/Specificity	3	[59,93,114]
ROC/AUC	10	[34,50,55,71,72,77,83,90,107,115]
TPR/Recall/	49	[33,35–42,45–50,52,53,60,65–67,69–72,74–76,79–83,85–

Sensitivity/DR		87,89,91,93,94, 97,98,100–102,105–115,117,118]
Precision	11	[41,43,59,65,66,76,80,89,113–115]
Accuracy	24	[35,36,41,47,48,50,53,59,60,64,66,75,80,81,88,89,92,99,100,103, 105, 113,114,116]
FPR / FAR	61	[32,33,35,36,38–46,48–50,52,55,56,58,60,62-67,69,71,72,74,75,79,81–83,85]

Conclusion

The main objective of our research is to study various papers related to web attacks and the techniques used for anomaly detection. One of the primary limitations identified in this systematic review is the absence of a standardized, up-to-date, and properly labelled dataset that enables the verification of experimental results acquired in various investigations. It is concerning that only 29.55 percent of the investigational results; found in the reviewed studies are dependent on publicly available datasets, with approximately half of these being heavily criticised by the scientific community, implying that additional research efforts are required to enable the creation and validation of a publicly available dataset. Additionally, a modest number of research utilising dimensionality reduction strategies have been identified. If PCA is utilised, it is recommended to use robust approaches, as the PCA method is extremely sensitive to outliers. If one of the variables in an observation is abnormal, the variance in this direction will be unnecessarily high. Due to the fact that PCA attempts to locate the paths with the greatest variance, the resulting subspace will have been excessively directed in this direction. The majority of the papers evaluated employ K-means and GMM classification algorithms in conjunction with Markov and SVM type models. It is usual to combine two or more clustering and/or classification techniques. Classic measures are commonly employed in studies relating to vulnerability identification. While these latter measurements may be useful, the authors feel that additional research should be undertaken in this area to develop a clear technique that enables the selection of certain metrics.

After reviewing the research, it was discovered that the majority of them do not define the type of assault they are attempting to prevent. So the prevention and detection of various attacks are the main objective of the work. In general, the various research papers that fit in deep learning methods demonstrate strong statistical performance; as a outcome, it was observed that the results vary depending upon the nature and type of the datasets. So, to promote the creation and usage of ; datasets for replicating and validating trials, it should be investigated further to determine whether deep learning truly enhances intrusion detection systems.

We discovered that deep learning algorithms outperform data mining; and the machine learning methodologies. Numerous deep learning methodologies have been investigated, however there is considerable room for improvement by tweaking hyper parameters and utilising alternative deep learning models. Logs are frequently utilised to detect anomalies in today's large-scale networks. However, due to massive log size expansion, traditional anomaly identification methods, which rely heavily on manual log examination, have become unworkable. However, inventors are still unaware of anomaly finding techniques, and occasionally, as a result of a lack of comprehensive examination and comparison of existing methodologies, they are forced to re-design an anomaly detection system.

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] Meena Siwach, Suman Mann, Anomaly detection for web log data : A Survey,IEEE Conference,2022.
[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.; Novovi˘cová, 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] Suman Mann, Deepa Gupta, Yukti Arora, Shivanka Priyanka Chugh, Akash Gupta, Smart Hospitals Using Artificial Intelligence and Internet of Things for COVID-19 Pandemic,chapter in Smart Healthcare Monitoring Using IoT with 5G, 2021Hu

[31] García-Teodoro, P.; Díaz-Verdejo, J.; Maciá-Fernández, G.; Vázquez, E. Anomaly-based network intrusion detection: Techniques, systems and challenges. Comput. Secur. 2009, 28, 18–28. [CrossRef]

[32] Thang, T.M.; Nguyen, K.V. FDDA: A Framework For Fast Detecting Source Attack In Web Application DDoS Attack. In Proceedings of the Eighth International Symposium on Information and Communication Technology, Nha Trang, Vietnam, 7–8 December 2017; Association for Computing Machinery: New York, NY, USA, 2017; SoICT 2017; pp. 278–285.

[33] Tripathi, N.; Hubballi, N. Slow Rate Denial of Service Attacks against HTTP/2 and Detection. Comput. Secur. 2018, 72, 255–272.

[34] Najafabadi, M.M.; Khoshgoftaar, T.M.; Calvert, C.; Kemp, C. User Behavior Anomaly Detection for Application Layer DDoS Attacks. In Proceedings of the 2017 IEEE International Conference on Information Reuse and Integration (IRI), San Diego, CA, USA, 4–6 August 2017; pp. 154–161.

[35] Zolotukhin, M.; Hämäläinen, T.; Kokkonen, T.; Siltanen, J. Increasing web service availability by detecting application-layer DDoS attacks in encrypted traffic. In Proceedings of the 2016 23rd International Conference on Telecommunications (ICT), Thessaloniki, Greece, 16–18 May 2016; pp. 1–6.

[36] Shirani, P.; Azgomi, M.A.; Alrabaee, S. A method for intrusion detection in web services based on time series. In Proceedings of the 2015 IEEE 28th Canadian Conference on Electrical and Computer Engineering (CCECE), Halifax, NS, Canada, 3–6 May 2015; pp. 836–841.

[37] Tripathi, N.; Hubballi, N.; Singh, Y. How Secure areWeb Servers? An Empirical Study of Slow HTTP DoS Attacks and Detection. In Proceedings of the 2016 11th International Conference on Availability, Reliability and Security (ARES), Salzburg, Austria, 31 August–2 September 2016; pp. 454–463.

[38] Wang, C.; Miu, T.T.N.; Luo, X.;Wang, J. SkyShield: A Sketch-Based Defense System Against Application Layer DDoS Attacks. IEEE Trans. Inf. Forensics Secur. 2018, 13, 559–573.

[39] Wang, Y.; Liu, L.; Si, C.; Sun, B. A novel approach for countering application layer DDoS attacks. In Proceedings of the 2017 IEEE 2nd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), Chongqing, China, 25–26 March 2017; pp. 1814–1817.

[40] Xie, Y.; Tang, S. Online Anomaly Detection Based on Web Usage Mining. In Proceedings of the 2012 IEEE 26th International Parallel and Distributed Processing SymposiumWorkshops PhD Forum, Shanghai, China, 21–25 May 2012; pp. 1177–1182.

[41] Lin, H.; Cao, S.; Wu, J.; Cao, Z.; Wang, F. Identifying Application-Layer DDoS Attacks Based on Request Rhythm Matrices. IEEE Access 2019, 7, 164480–164491.

[42] Xiao, R.; Su, J.; Du, X.; Jiang, J.; Lin, X.; Lin, L. SFAD: Toward effective anomaly detection based on session feature similarity. Knowl.-Based Syst. 2019, 165, 149–156.

[43] Kozik, R.; Chora´s, M.; Hołubowicz,W. Evolutionary-based packets classification for anomaly detection in web layer. Secur. Commun. Netw. 2016, 9, 2901–2910.

[44] Wang, L.; Cao, S.; Wan, L.; Wang, F. Web Anomaly Detection Based on Frequent Closed Episode Rules. In Proceedings of the 2017 IEEE Trustcom/BigDataSE/ICESS, Sydney, NSW, Australia, 1–4 August 2017; pp. 967–972.

[45] Yuan, G.; Li, B.; Yao, Y.; Zhang, S. A deep learning enabled subspace spectral ensemble clustering approach for web anomaly detection. In Proceedings of the 2017 International Joint Conference on Neural Networks (IJCNN), Anchorage, AK, USA, 14–19 May 2017; pp. 3896–3903.

[46] Bronte, R.; Shahriar, H.; Haddad, H. Information Theoretic Anomaly Detection Framework for Web Application. In Proceedings of the 2016 IEEE 40th Annual Computer Software and Applications Conference (COMPSAC), Atlanta, GA, USA, 10–14 June 2016; Volume 2, pp. 394–399. [CrossRef]

[47] Luo, Y.; Cheng, S.; Liu, C.; Jiang, F. PU Learning in Payload-based Web Anomaly Detection. In Proceedings of the 2018 Third International Conference on Security of Smart Cities, Industrial Control System and Communications (SSIC), Shanghai, China, 18–19 October 2018; pp. 1–5. [CrossRef]

[48] Ren, X.; Hu, Y.; Kuang, W.; Souleymanou, M.B. A Web Attack Detection Technology Based on Bag of Words and Hidden Markov Model. In Proceedings of the 2018 IEEE 15th International Conference on Mobile Ad Hoc and Sensor Systems (MASS), Chengdu, China, 9–12 October 2018; pp. 526–531.

[49] Kozik, R.; Chora´s, M.; Hołubowicz,W. HardeningWeb Applications against SQL Injection Attacks Using Anomaly Detection Approach. In Image Processing & Communications Challenges 6; Chora´s, R.S., Ed.; Springer International Publishing: Cham, Switzerland, 2015; pp. 285–292.

[50] Maggi, F.; Robertson,W.; Kruegel, C.; Vigna,G. Protecting aMoving Target: Addressing Web Application Concept Drift. In Recent Advances in Intrusion Detection; Kirda, E., Jha, S., Balzarotti, D., Eds.; Springer:Berlin/Heidelberg, Germany, 2009; pp. 21–40.

[51] Valeur, F.; Vigna, G.; Kruegel, C.; Kirda, E. An Anomaly-Driven Reverse Proxy for Web Applications. In Proceedings of the 2006 ACM Symposium on Applied Computing, Dijon, France, 23–27 April 2006; Association for Computing Machinery: New York, NY, USA, 2006; pp. 361–368.

[52] Guangmin, L. Modeling Unknown Web Attacks in Network Anomaly Detection. In Proceedings of the 2008 Third International Conference on Convergence and Hybrid Information Technology, Busan, Korea, 11–13 November 2008; Volume 2, pp. 112–116.

[53] Yu, S.; Guo, S.; Stojmenovic, I. Fool Me If You Can: Mimicking Attacks and Anti-Attacks in Cyberspace. IEEE Trans. Comput. 2015, 64, 139–151.

[54] Sakib, M.N.; Huang, C. Using anomaly detection based techniques to detect HTTP-based botnet C C traffic. In Proceedings of the 2016 IEEE International Conference on Communications (ICC), Kuala Lumpur, Malaysia, 22–27 May 2016; pp. 1–6.

[55] Medvet, E.; Bartoli, A. On the Effects of Learning Set Corruption in Anomaly-Based Detection of Web Defacements. In Detection of Intrusions and Malware, and Vulnerability Assessment; Hämmerli, M.B., Sommer, R., Eds.; Springer: Berlin/Heidelberg, Germany, 2007; pp. 60–78.

[56] Davanzo, G.; Medvet, E.; Bartoli, A. Anomaly detection techniques for a web defacement monitoring service. Expert Syst. Appl. 2011, 38, 12521–12530.

[57] Juvonen, A.; Sipola, T.; Hämäläinen, T. Online anomaly detection using dimensionality reduction techniques for HTTP log analysis. Comput. Netw. 2015, 91, 46–56.

[58] Wang,W.; Guyet, T.; Quiniou, R.; Cordier, M.O.; Masseglia, F.; Zhang, X. Autonomic Intrusion Detection. Know.-Based Syst. 2014, 70, 103–117.

[59] Vartouni, A.M.; Kashi, S.S.; Teshnehlab, M. An anomaly detection method to detect web attacks using Stacked Auto-Encoder. In Proceedings of the 2018 6th Iranian Joint Congress on Fuzzy and Intelligent Systems (CFIS), Kerman, Iran, 28 February–2 March 2018; pp. 131–134.

[60] Zolotukhin, M.; Hämäläinen, T.; Kokkonen, T.; Siltanen, J. Analysis of HTTP requests for anomaly detection of web attacks. In Proceedings of the 2014 World Ubiquitous Science Congress: 2014 IEEE 12th International Conference on Dependable, Autonomic and Secure Computing, DASC 2014, Dalian, China, 24–27 August 2014; IEEE: Piscataway, NJ, USA, 2014; pp. 406–411.

[61] Asselin, E.; Aguilar-Melchor, C.; Jakllari, G. Anomaly detection for web server log reduction: A simple yet efficient crawling based approach. In Proceedings of the 2016 IEEE Conference on Communications and Network Security (CNS), Philadelphia, PA, USA, 17–19 October 2016; pp. 586–590.

[62] Zhang, S.; Li, B.; Li, J.; Zhang, M.; Chen, Y. A Novel Anomaly Detection Approach for Mitigating Web-Based Attacks Against Clouds. In Proceedings of the 2015 IEEE 2nd International Conference on Cyber Security and Cloud Computing (CSCloud), New York, NY, USA, 3–5 November 2015; IEEE Computer Society: Piscataway, NJ, USA, 2015; pp. 289–294.

[63] Zhang, M.; Lu, S.; Xu, B. An Anomaly Detection Method Based on Multi-models to Detect Web Attacks. In Proceedings of the 2017 10th International Symposium on Computational Intelligence and Design (ISCID), Hangzhou, China, 9–10 December 2017; Volume 2, pp. 404–409.

[64] Parhizkar, E.; Abadi, M. OC-WAD: A one-class classifier ensemble approach for anomaly detection in web traffic. In Proceedings of the 2015 23rd Iranian Conference on Electrical Engineering, Tehran, Iran, 10–14 May 2015; pp. 631–636.

[65] Kozik, R.; Choras, M. Adapting an Ensemble of One-Class Classifiers for aWeb-Layer Anomaly Detection System. In Proceedings of the 2015 10th International Conference on P2P, Parallel, Grid, Cloud and Internet Computing, 3PGCIC, Krakow, Poland, 4–6 November 2015; IEEE: Piscataway, NJ, USA, 2015; pp. 724–729.

[66] Cao, Q.; Qiao, Y.; Lyu, Z. Machine learning to detect anomalies in web log analysis. In Proceedings of the 2017 3rd IEEE International Conference on Computer and Communications (ICCC), Chengdu, China, 13–16 December 2017; pp. 519–523.

[67] Yu, J.; Tao, D.; Lin, Z. A hybrid web log based intrusion detection model. In Proceedings of the 2016 4th IEEE International Conference on Cloud Computing and Intelligence Systems, CCIS 2016, Beijing, China, 17–19 August 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 356–360.

[68] Threepak, T.;Watcharapupong, A. Web attack detection using entropy-based analysis. In Proceedings of the International Conference on Information Networking, Phuket, Thailand, 10–12 February 2014; IEEE: Piscataway, NJ, USA, 2014; pp. 244–247.

[69] Swarnkar, M.; Hubballi, N. Rangegram: A novel payload based anomaly detection technique against web traffic. In Proceedings of the 2015 IEEE International Conference on Advanced Networks and Telecommuncations Systems (ANTS), Kolkata, India, 15–18 Decemnber 2015; pp. 1–6.

[70] Xu, H.; Tao, L.; Lin, W.; Wu, Y.; Liu, J.; Wang, C. A model for website anomaly detection based on log analysis. In Proceedings of the 2014 IEEE 3rd International Conference on Cloud Computing and Intelligence Systems, Shenzhen, China, 27–29 November 2014; pp. 604–608.

[71] Park, S.; Kim, M.; Lee, S. Anomaly Detection for HTTP Using Convolutional Autoencoders. IEEE Access 2018, 6, 70884–70901.

[72] Chora´s, M.; Kozik, R. Machine learning techniques applied to detect cyberattacks on web applications. Log. J. IGPL 2014, 23, 45–56.

[73] Tharshini, M.; Ragavinodini, M.; Senthilkumar, R. Access Log Anomaly Detection. In Proceedings of the 2017 Ninth International Conference on Advanced Computing (ICoAC), Chennai, India, 14–16 December 2017; pp. 375–381.

[74] Kozik, R.; Chora´s, M.; Hołubowicz,W. Packets tokenization methods for web layer cyber security. Log. J. IGPL **2016**, 25, 103–113.

[75] Kamarudin, M.H.; Maple, C.; Watson, T.; Safa, N.S. A LogitBoost-Based Algorithm for Detecting Known and UnknownWeb Attacks. IEEE Access **2017**, 5, 26190–26200.

[76] Yu, Y.; Liu, G.; Yan, H.; Li, H.; Guan, H. Attention-Based Bi-LSTM Model for Anomalous HTTP Traffic Detection. In Proceedings of the 2018 15th International Conference on Service Systems and Service Management (ICSSSM), Hangzhou, China, 21–22 July 2018; pp. 1–6.

[77] Nguyen, X.N.; Nguyen, D.T.; Vu, L.H. POCAD: A novel pay load-based one-class classifier for anomaly detection. In Proceedings of the 2016 3rd National Foundation for Science and Technology Development Conference on Information and Computer Science (NICS), Danang, Vietnam, 14–16 September 2016; pp. 74–79.

[78] Lu, L.; Zhu, X.; Zhang, X.; Liu, J.; Bhuiyan, M.Z.A.; Cui, G. One Intrusion Detection Method Based On Uniformed Conditional Dynamic Mutual Information. In Proceedings of the 2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/ 12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE), New York, NY, USA , 1–3 August 2018; pp. 1236–1241.

[79] Moustafa, N.; Misra, G.; Slay, J. Generalized Outlier Gaussian Mixture technique based on Automated Association Features for Simulating and DetectingWeb Application Attacks. IEEE Trans. Sustain. Comput. **2018**, 1.

[80] Alrawashdeh, K.; Purdy, C. Fast Activation Function Approach for Deep Learning Based Online Anomaly Intrusion Detection. In Proceedings of the 2018 IEEE 4th International Conference on Big Data Security on Cloud (BigDataSecurity), IEEE International Conference on High Performance and Smart Computing, (HPSC) and IEEE International Conference on Intelligent Data and Security (IDS), Omaha, NE, USA, 3–5 May 2018; pp. 5–13.

[81] Kaur, R.; Bansal, M. Multidimensional attacks classification based on genetic algorithm and SVM. In Proceedings of the 2016 2nd International Conference on Next Generation Computing Technologies (NGCT), Dehradun, India, 14–16 October 2016; pp. 561–565.

[82] Angiulli, F.; Argento, L.; Furfaro, A. Exploiting N-Gram Location for Intrusion Detection. In Proceedings of the 2015 IEEE 27th International Conference on Tools with Artificial Intelligence (ICTAI), Vietri sul Mare, Italy, 9–11 November 2015; pp. 1093–1098.

[83] Hiremagalore, S.; Barbará, D.; Fleck, D.; Powell, W.; Stavrou, A. transAD: An Anomaly Detection Network Intrusion Sensor for the Web. In Information Security; Chow, S.S.M., Camenisch, J., Hui, L.C.K., Yiu, S.M., Eds.; Springer International Publishing: Cham, Switzerland, 2014; pp. 477–489.

[84] Favaretto, M.; Spolaor, R.; Conti, M.; Ferrante, M. You Surf so Strange Today: Anomaly Detection in Web Services via HMM and CTMC. In Green, Pervasive, and Cloud Computing; Au, M.H.A., Castiglione, A., Choo, K.K.R.;, Palmieri, F., Li, K.C., Eds.; Springer International Publishing: Cham, Switzerland, 2017; pp. 426–440.

[85] Kozik, R.; Chorás, M. The http content segmentation method combined with adaboost classifier for web-layer anomaly detection system. Adv. Intell. Syst. Comput. **2017**, 527, 555–563.

[86] Kozik, R.; Chora´s, M.; Holubowicz, W.; Renk, R. Extreme Learning Machines for Web Layer Anomaly Detection. In Image Processing and Communications Challenges 8; Chora´s, R.S., Ed.; Springer International Publishing: Cham, Switzerland, 2017; pp. 226–233.

[87] Kozik, R.; Chora´s, M.; Renk, R.; Holubowicz, W. Patterns Extraction Method for Anomaly Detection in HTTP Traffic. In Proceedings of the International Joint Conference; Herrero, Á., Baruque, B., Sedano, J., Quintián, H., Corchado, E., Eds.; Springer International Publishing: Cham, Switzerland, 2015; pp. 227–236.

[88] Shi, Y.; Wang, S.; Zhao, Q.; Li, J. A Hybrid Approach of HTTP Anomaly Detection. In Web and Big Data; Springer International Publishing: Cham, Switzerland, 2017; pp. 128–137.

[89] Kim, T.Y.; Cho, S.B. Web traffic anomaly detection using C-LSTM neural networks. Expert Syst. Appl. 2018, 106, 66–76..

[90] Jin, X.; Cui, B.; Li, D.; Cheng, Z.; Yin, C. An improved payload-based anomaly detector for web applications. J. Netw. Comput. Appl. 2018, 106, 111–116.

[91] Wang, W.; Liu, J.; Pitsilis, G.; Zhang, X. Abstracting massive data for lightweight intrusion detection in computer networks. Inf. Sci. 2018, 433–434, 417–430.

[92] Liu, T.; Zhang, L. Application of Logistic Regression in WEB Vulnerability Scanning. In Proceedings of the 2018 International Conference on Sensor Networks and Signal Processing (SNSP), Xi'an, China, 28–31 October 2018; pp. 486–490.

[93] Betarte, G.; Gimenez, E.; Martinez, R.; Pardo, A. ImprovingWeb Application Firewalls through Anomaly Detection. In Proceedings of the 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA), Orlando, FL, USA, 17–20 December 2018; pp. 779–784.

[94] Li, B.; Yuan, G.; Shen, L.; Zhang, R.; Yao, Y. Incorporating URL embedding into ensemble clustering to detect web anomalies. Future Gener. Comput. Syst. 2019, 96, 176–184.

[95] Yun, Y.; Park, S.; Kim, Y.; Ryou, J. A Design and Implementation of Profile Based Web Application Securing Proxy. In Information Security Practice and Experience; Chen, K., Deng, R., Lai, X., Zhou, J., Eds.; Springer: Berlin/Heidelberg, Germany, 2006; pp. 248–259.

[96] Bolzoni, D.; Etalle, S. Boosting Web Intrusion Detection Systems by Inferring Positive Signatures. In On the Move to Meaningful Internet Systems: OTM 2008; Meersman, R., Tari, Z., Eds.; Springer: Berlin/Heidelberg, Germany, 2008; pp. 938–955.

[97] Li, Y.; Guo, L.; Tian, Z.H.; Lu, T.B. A Lightweight Web Server Anomaly Detection Method Based on Transductive Scheme and Genetic Algorithms. Comput. Commun. 2008, 31, 4018–4025.

[98] Kruegel, C.; Vigna, G.; Robertson,W. A multi-model approach to the detection of web-based attacks. Comput. Netw. 2005, 48, 717–738.

[99] Cho, S.; Cha, S. SAD:Web session anomaly detection based on parameter estimation. Comput. Secur. 2004, 23, 312–319.

[100] Yamada, A.; Miyake, Y.; Takemori, K.; Studer, A.; Perrig, A. Intrusion Detection for Encrypted Web Accesses. In Proceedings of the 21st International Conference on Advanced Information Networking and Applications Workshops (AINAW'07), Niagara Falls, ON, Canada, 21–23 May 2007; Volume 1, pp. 569–576.

[101] Yan, C.; Qin, Z.; Shi, Y. Sequence Analysis and Anomaly Detection of Web Service Composition. In Proceedings of the 2008 International Conference on Computer Science and Software Engineering, Wuhan, China, 12–14 December 2008; Volume 3; pp. 1043–1048.

[102] Jamdagni, A.; Tan, Z.; Nanda, P.; He, X.; Liu, R.P. Intrusion Detection Using GSAD Model for HTTP Traffic on Web Services. In Proceedings of the 6th International Wireless Communications and Mobile Computing Conference, Caen, France, 15 January 2010; Association for Computing Machinery: New York, NY, USA; pp. 1193–1197.

[103] Wang,W.; Zhang, X. High-SpeedWeb Attack Detection through Extracting Exemplars from HTTP Traffic. In Proceedings of the 2011 ACM Symposium on Applied Computing, TaiChung, Taiwan, 21–24 March 2011; Association for Computing Machinery: New York, NY, USA; pp. 1538–1543.

[104] Kruegel, C.; Vigna, G. Anomaly detection of Web-based attacks. In Proceedings of the ACM Conference on Computer and Communications Security,Washington, DC, USA, 27–30 October 2003; ACM Press: New York, NY, USA; pp. 251–261.

[105] Rahnavard, G.; Najjar, M.S.A.; Taherifar, S. A method to evaluateWeb Services Anomaly Detection using Hidden Markov Models. In Proceedings of the 2010 International Conference on Computer Applications and Industrial Electronics, Kuala Lumpur, Malaysia, 5–8 December 2010; pp. 261–265.

[106] Das, D.; Sharma, U.; Bhattacharyya, D.K. AWeb Intrusion Detection Mechanism based on Feature based Data Clustering. In Proceedings of the 2009 IEEE International Advance Computing Conference, Patiala, India, 6–7 March 2009; pp. 1124–1129.

[107] Li, X.; Xue, Y.; Malin, B. Detecting Anomalous User Behaviors in Workflow-Driven Web Applications. In Proceedings of the 2012 IEEE 31st Symposium on Reliable Distributed Systems, Irvine, CA, USA, 8–11 October 2012; pp. 1–10.

[108] Le, M.; Stavrou, A.; Kang, B.B. DoubleGuard: Detecting Intrusions in Multitier Web Applications. IEEE Trans. Dependable Secur. Comput. 2012, 9, 512–525.

[109] Xie, Y.; Yu, S.Z. Light-weight detection of HTTP attacks for large-scale Web sites. In Proceedings of the 2008 11th IEEE Singapore International Conference on Communication Systems, Guangzhou, China, 19–21 November 2008; pp. 1182–1186.

[110] Sriraghavan, R.G.; Lucchese, L. Data processing and anomaly detection in web-based applications. In Proceedings of the 2008 IEEE Workshop on Machine Learning for Signal Processing, Cancun, Mexico, 16–19 October 2008; pp. 187–192.

[111] Fan, W.K.G. An adaptive anomaly detection of WEB-based attacks. In Proceedings of the 2012 7th International Conference on Computer Science Education (ICCSE), Melbourne, VIC, Australia, 14–17 July , 2012; pp. 690–694.

[112] Kirchner, M. A framework for detecting anomalies in HTTP traffic using instance-based learning and k-nearest neighbor classification. In Proceedings of the 2010 2nd International Workshop on Security and Communication Networks (IWSCN), Karlstad, Sweden, 26–28 May 2010; pp. 1–8.

[113] Kakavand, M.; Mustapha, A.; Tan, Z.; Yazdani, S.F.; Arulsamy, L. O-ADPI: Online Adaptive Deep-Packet Inspector Using Mahalanobis Distance Map for Web Service Attacks Classification. IEEE Access 2019, 7, 167141–167156.

[114] Moradi Vartouni, A.; Teshnehlab, M.; Sedighian Kashi, S. Leveraging deep neural networks for anomaly-based web application firewall. IET Inf. Secur. 2019, 13, 352–361.

[115] Li, J.; Fu, Y.; Xu, J.; Ren, C.; Xiang, X.; Guo, J. Web application attack detection based on attention and gated convolution networks. IEEE Access 2019.

[116] Alhakami, W.; ALharbi, A.; Bourouis, S.; Alroobaea, R.; Bouguila, N. Network Anomaly Intrusion Detection Using a Nonparametric Bayesian Approach and Feature Selection. IEEE Access 2019, 7, 52181–52190.

[117] Kozik, R.; Chora´s, M. Protecting the application layer in the public domain with machine learning methods. Log. J. IGPL 2018, 27, 149–159.

[118] Jin, L.;Wang, X.J.; Zhang, Y.; Yao, L. Anomaly Detection in theWeb Logs Using Unsupervised Algorithm. In Human Centered Computing; Tang, Y., Zu, Q., Rodríguez García, J.G., Eds.; Springer International Publishing: Cham, Switzerland, 2019; pp. 393–405.

[119] Bhattacharyya, D.K.; Kalita, J.K. DDoS Attacks: Evolution, Detection, Prevention, Reaction, and Tolerance; CRC Press: Boca Raton, FL, USA, 2016.

[120] Meena Siwach, Suman Mann, Anomaly detection for web log data analysis using improved PCA Technique, Journal of information and optimization Science. 131-141, DOI: 10.1080/02522667.2022.2037283
Injection_Flaws (accesed on 15 May 2020).

[121] Wei, K.; Muthuprasanna, M.; Kothari, S. Preventing SQL injection attacks in stored procedures.
In Proceedings of the Australian Software Engineering Conference (ASWEC'06), Sydney, NSW, Australia, 18–21 April 2006; pp. 8–198.

[122] Leonard, J.; Xu, S.; Sandhu, R. A Framework for Understanding Botnets. In Proceedings of the 2009 International Conference on Availability, Reliability and Security, Fukuoka, Japan, 16–19 March 2009; pp. 917–922.

[123] Suman Mann, Archana Balyan, Vinita Rohilla, Deepa Gupta, Zatin Gupta, Abdul Wahab Rahmani, "Artificial Intelligence-based Blockchain Technology for Skin Cancer Investigation Complemented with Dietary Assessment and Recommendation using Correlation Analysis in Elder Individuals", *Journal of Food Quality*, vol. 2022, Article ID 3958596, 7 pages, 2022. https://doi.org/10.1155/2022/3958596

[124] Hadianto, R.; Purboyo, T.W. A Survey Paper on Botnet Attacks and Defenses in Software Defined Networking. Int. J. Appl. Eng. Res. 2018, 13, 483–489.

[125] D. Gupta, S. K. Jha and S. Mann Maharaja Surajmal, "Internet Crimes-It's Analysis and Prevention Approaches," 2021 9th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), 2021, pp. 1-4, doi: 10.1109/ICRITO51393.2021.9596396