International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys. 10, No.1 (Jann-2021)

http://dx.doi.org/10.12785/ijcds/100111

A Comparative Review of Malware Analysis and Detection in HTTPs Traffic

Abhay Pratap Singh [1] and Mahendra Singh [2]

[1, 2] Department of Computer Science, Gurukula Kangri Vishwavidyalaya Haridwar, India

Received 3 Apr.2020, Revised 30 Jul. 2020, Accepted 13 Nov. 2020, Published 1 Jul. 2021

Abstract: HTTPs is essentially an integration of the Hypertext Transfer Protocol with either TLS or SSL. The responsibility of SSL/TLS in HTTPs is to encrypt the content of HTTP. Without encryption, the communication can be comprehended by anyone that keeps up seeing the packets between the sender and receiver. As a higher amount of web traffic shifts towards encrypted traffic, concealing an attack in encrypted communication will develop in prominence and refinement. Malware poses one of the significant digital security risks in the present scenario, with the goal of malware is to exfiltrate information from networks and misusing it. The measure of malwares utilizing HTTPs traffic for their communication is on the rise year by year. This situation is obscure to handle for cyber security researchers because malware traffic is encrypted, and it primarily looks like regular traffic. The detection and analysis of malware in HTTPs traffic is challenging because application data is encrypted between the client and server. This paper endeavors to analytically review the concepts and techniques for malware analysis and detection in HTTPs traffic and performs a comparative study of state of the art. The review suggests that most of the techniques are using the statistical features of network traffic and machine-learning based techniques in order to detect and classify malware in encrypted traffic.

Keywords: Malware, Botnet, Encryption, Network Security, SSL/TLS

1. INTRODUCTION

The HTTPs (hypertext transfer protocol secure) protocol is a standout amongst the most well-known protocol in computer network organization that gives a protected communication between networks. HTTPs is a combination of HTTP and SSL/TLS. As per a Google report [1] of April 2017, the use of HTTPs is on the rise. The report demonstrates that PC users download more than 50% of the web pages using HTTPs, and utilized 66% of their time in HTTPs pages. With this growing usage of encrypted network traffic on the whole internet, malware has also begun to utilize the HTTPs to secure its own communication. The diversity of encrypted malware or encoded malware is increasing, and attackers are also utilizing different techniques to convey malware like code obfuscation, drive-by downloads, encryption, etc. Unfortunately, encryption is a twofold edged sword, while genuine clients utilize encryption for all the genuine reason; the cyber attackers utilize this to avoid detection and secure their malicious activities. Malware protection for a computer system is one of the utmost network security tasks for individual users and businesses because even a single cyber-attack can result in data leakage and adequate losses. The significant losses and frequent types of cyber-attack point out the need for precise and timely detection methods. The growing volume of encrypted web traffic, both genuine and malicious, poses much more difficulties and perplexity for protectors endeavoring to recognize and monitor potential threats. Encryption is a means to update security in many ways; however, it likewise gives malicious actors a vast apparatus to cover command – and – control (otherwise called c2and CC) activity, managing them enough time to work and inflict damage. The identification of HTTPs malware traffic is challenging and complex on the grounds that the communication is encrypted between the client and server that give a favorable position to the attacker to set up malware. Generally, network security tools neglect to recognize this sort of threat. The common solution for managing and inspecting HTTPs traffic in big companies is to introduce HTTPs interceptor proxies. This interceptor is set between the client and server. The enciphered traffic is deciphered, examined whether it contains malicious traffic or not, encrypted again and sent to the destination IP (internet address) as shown in Figure 1.

E-mail: rs.abhaypratapsingh@gkv.ac.in, msa@gkv.ac.in

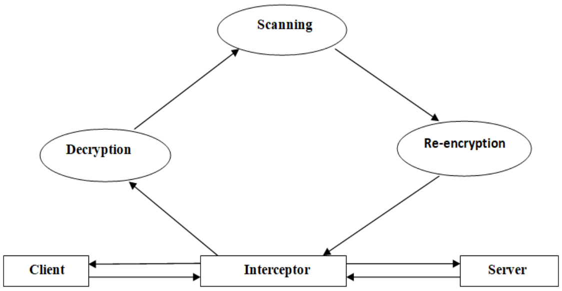

Figure 1. HTTPs Interception

This is a regular methodology that takes into consideration and traditional recognition techniques to be utilized for detecting decrypted malicious traffic. One of the drawbacks of using a HTTPs interceptor between the client and server is that it violates the fundamental concept of HTTPs, which is having a secret, safe and secure communications. This paper reviews several concepts, techniques proposed, used, and practiced for malware analysis and detection. The contributions of the paper can be put in the following way:

- Overview of SSL/TLS encryption mechanism.
- Deep insight into malware analysis and evolution of encrypted malware.
- Explains attack vectors in encrypted traffic
- Review and comparative study of various existing techniques and models employed to detect malware in HTTPs traffic.

This paper is organized as follows. Section 2 reviews the background of SSL/TLS encryption. In Section 3 malware analysis types are explored. The evolution of encrypted malware and encoding/decoding techniques are discussed in section 4. Section 5 describes various attack vectors used in encrypted traffic. Section 6 reviews and compares the state of art for malware analysis and detection in HTTPs traffic. Different evaluation metrics are explained in Section 7. The paper is summarized with future scope of investigation in Section 8.

2. BACKGROUND

The SSL/TLS protocol facilitates the encryption of HTTPs communication. To start with, an initial handshake process is performed between the client and server. In the process of initial handshake, cryptographic parameters are exchanged, such as cipher suites, version number, digital certificate, etc. SSL and TLS are the protocols used for the encipher of network traffic. These protocols are also used to maintain integrity, authentication, and confidentiality for data in transmission. We contribute necessary information to ease understanding the rest of our paper.

A. *Traffic Encryption with SSL/TLS*

The TLS is replacing SSL as the security mechanism to encode the transmission between internal web browsers and web servers [2, 3]; thus, TLS is the successor to SSL. The SSL is no longer utterly secure in today's environments. TLS v 1.2 is the most current version in client browsers and web servers, even though TLS v 1.3 has been launched, but it's not mostly used in client browser and server. It is fundamentally utilized for anchoring HTTP, FTP, SMTP sessions, and for Virtual Private Networks or VoIP (voice over internet protocol). The TLS comprises two different protocols, the TLS record protocol and the TLS handshake protocol, and both are characterized in RFC (Request For Comments) 5246 [4]. The TLS record protocol utilizes symmetric key encryption and needs a dependable connection, for example, that given by TCP. The symmetric key for the TLS record protocol is commonly consulted by the TLS handshake protocol [4]. The TLS handshake protocol has three essential properties: the endpoint's identity can be validated through exchanging the proper cryptographic key distribution in the handshake process is secure, inaccessible through spying or Man-In-the-Middle (MITM) attacks; and no outsider can alter the transaction without changing the original endpoints [4]. The prime objective of TLS is to encapsulate other traffic layers, for example, HTTP.

The TLS handshake protocol included the following stages recorded underneath from RFC 5246 [4] as depicted in Figure 2 also:

- Exchange hello message to agree with algorithms, swap random values, and inspect session regeneration between client and server.
- Send and receive the required encryption arguments to enable the client and server to utilize a pre-master secret.
- Exchange digital certificates and encrypted information to enable the client and servers to authenticate them.
- Produce a master key from the pre-master secret and exchange random values.
- Facilitate protection for the features of the record layer.
- Enable the client and server to validate that corresponding peer has computed similar protection features and that the SSL/TLS handshake happened without altering by an attacker.

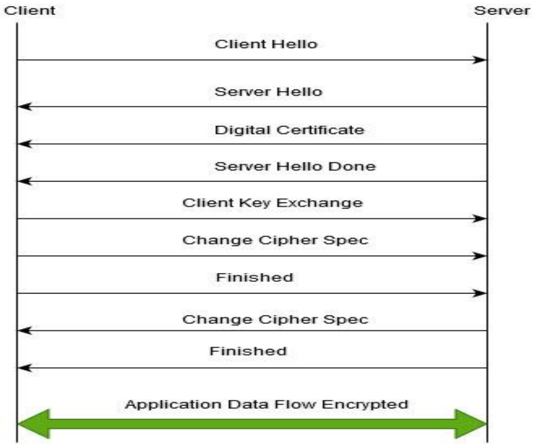

Figure 2. SSL/TLS Handshake Process

The TLS handshake protocol incorporates the choice for including extensions to the hello messages of client [4, 5]. These extensions are needed by the protocol to open the connection. One such extension is a server name, which is particularly valuable when a single physical host is running on various virtual servers [5]. The server name enables the physical host to react with the right digital certificate and start a connection with the coveted virtual server. Currently, an ever-growing number of applications are being moved to the cloud and utilizing internet based services [6, 7]. These web-based services regularly utilize encrypted network traffic protocols, for example, SSL/TLS [8, 9].

3. Types of Malware Analysis

Malware analysis is a systematic way toward determining the purpose and function of a known malware sample. The fundamental objective of malware analysis is to understand how the malware behaves, how to identify malware and remove it. The various types of malware analysis techniques are presented in Figure 3.

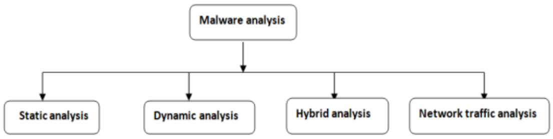

Figure 3. Types of Malware Analysis

A. Static analysis

Static analysis means that the malware analysis is completed without executing the code. This type of analysis is suitable as we do not need to execute the binary code; hence, it requires fewer amounts of resources and time. The binary codes are checked through disassembling the executable file. Nowadays, malware authors use various methods to prevent static analysis from detecting malicious code.

B. Dynamic analysis

In this analysis, malicious samples are executed and monitored like a virtual machine (VM), an emulator, or a simulator [10]. This type of analysis is also called as behavior analysis. When an executable file runs, we can track all kinds of relevant information related to malicious code. Dynamic analysis is more effective rather than compared to static analysis; furthermore, dynamic analysis is able to detect known and unknown malware [11]. Nevertheless, dynamic analysis is time-intensive and resource-consuming [12].

C. Hybrid analysis

Hybrid analysis is the integration of static and dynamic analysis. With the support of hybrid analysis, cyber security researchers get the benefits of both analysis, static and dynamic. Hence, it increases the ability to detect malicious programs correctly [13]. Both types of analysis have their own advantages and disadvantages. Static analysis is useful and faster compared to dynamic analysis. However, malware developers use various methods to bypass static analysis based engine. Contrarily, dynamic analysis can detect unknown and known malware.

D. Network traffic analysis

In order to inspect network traffic, generally, we used packet-based features and flow-based features to detect malware. The Packet based approach inspects the entire payload content besides headers. Packet based traffic analysis is an entirely passive approach, which means that it can provide much more information related to network problems. In [14], authors extract features on the basis of the size of packet to execute fingerprinting attacks on website against enciphered traffic. Flow- based features provide useful information related to network connection instead of the packet payload. The Flow based approach determines the network traffic statistical overview. A flow is well-defined as the identical source IP, destination IP, protocol, source port, destination port. Besides, flow based features are used to detect malware in network traffic; some other papers have also utilized these features in a more precise manner [15] [16].

4. The Evolution of Encrypted Malware

Over the years, malware has become a global threat to cyber security research community. It becomes more complicated when we add one more word encryption; the use of encryption is used by attackers increasing year by year. The growth of the encrypted malware has typically given attackers the perfect place to conceal encrypted malware, making it impossible to identify and detect malicious packets in a network [17]. The trend of attacks on encrypted traffic is depicted in Figure [4].

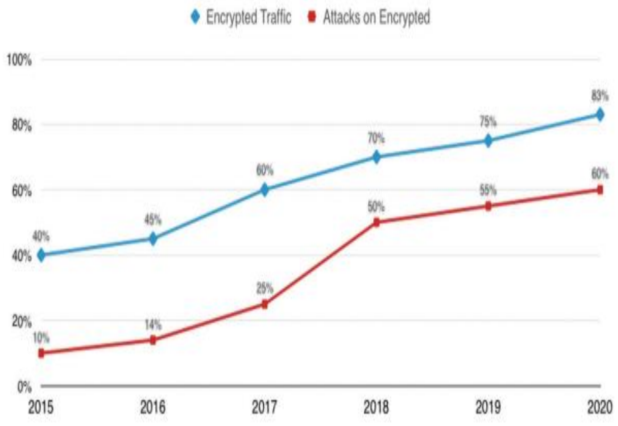

Figure 4. Growth of Encrypted Traffic and Attack [17]

A. Encoding and Decoding Techniques for Malware

Cyber attackers adopt various techniques to avoid their malicious codes detection against antivirus software and to further complicate the malware analysis process [18]. In this section, we explain different types of encoding and decoding for malware that is favorable choice for cyber attackers.

*1) **Encoding with base64:*** Base64 is an encoding technique designed to represent data in an ASCII text format [19]. It is often used to encode and decode malware, and base64 provides a significant advantage to cyber attackers while providing the least benefit to defenders. The prime objective of base 64 is to standardize code. Cyber attackers are also utilizing this approach by injecting false or gibberish character strings that mimic standard base 64, and code works well until its decode [20].

*2) **Code packing:*** Code packing is a kind of technique used to hide the code of a program through one or more layers of encryption/compression. It is a subset of techniques and tools that modifies a malicious software code in such a way that traditional based antimalware software unable to detect these harmful codes [21].

*3) **Polymorphic:*** Malware authors used this technique to conceal signature-based malware detection by doing small and internal changes in characteristics of the malicious code [22]. In this type of technique, malware authors used several complex encrypting algorithms that mutate themselves through self-encryption [23].

*4) **Metamorphic:*** Metamorphic is a superior variant of the polymorphic technique, where the complete internal structure is encoded [24]. The metamorphic technique is also known as “body polymorphic.” In this approach, it is continuously reprogramming the malicious code in each execution iteration/distribution to evade detection without altering the control flow [25].

*5) **A debugging approach:*** One of the most popular methods for malware analysis is disassemblers and debuggers to inspect the working of specific code. This type of mechanism is also called reverse engineering, where the malicious code is loaded into a disassembler such as IDA Pro [26]. Later, malware analysts can use a debugger to execute every code path, write a script and found some interesting pattern and decode the malicious code [27]. However, this type of approach is very tedious for every relevant code.

5. Attack Vectors in Encrypted Traffic

Currently, many malware developers have introduced their latest variants to use SSL/TLS to encrypt malware communications from infected clients. There are many ways by using which malware developers exploit encrypted traffic. A few of them are described below:

A. Drive-by download

A drive-By download is a type of software that installs itself without keeping the user aware of it. Consequently, drive-by downloads poses a significant menace to the internet and its users [28]. In this style of attack, merely accessing a website that contains malicious content may result in malware infecting the user’s computer. The malicious code, installed as part of the attack, then take over the victim host. To diminish the threats of drive-by downloads; keep web-browser and operating system up to date in regular intervals [29].

B. Phishing Email

Phishing is one of the supreme and conventional methods to send malware on victim machines. The attackers smartly design a legitimate email and send it to the victim. One of the best ways for an attacker to choose to spread malware is through a spear-phishing attack. In this style of attack, attackers usually collected all types of information about a specific person or companies then launch an attack. The possibilities of getting successful chances are more significant than other types of phishing attacks [30]. Some of the notorious forms of phishing attacks include Deceptive phishing, DNS-based phishing, and Search Engine phishing [31].

C. Malvertising Campaign

Malvertising is a new technique to spread malware with the help of online advertising. Online advertising provides an excellent platform for spreading malware because it is easy to attract users to sell or watch new products online. The amusing fact about infections spread through Malvertising is that it does not need any user-oriented action, such as clicking to compromise the system and does not exploit any vulnerabilities on the website or the server hosting it. Infections carried through Malvertising quietly travel through web page advertisements [32]. However, detecting such kinds of threats is challenging because of the rapid changes of ads on a website [33].

D. Exploit Kit

Exploit kit represents a kind of attacking toolkit which is used by malware developers to take advantage of vulnerabilities in a system so that they can deliver malware or do other malicious activities. Exploit kit is an infection kit that is utilized to perform malicious code onto the user's system. A popular way to spread malware through an exploit kit is that if users visit a website that is already hosted on an exploit kit, then malware could be downloaded automatically without user knowledge. Some of the popular types of exploit kits are Angler, Spelevo, Fallout, and Magnitude [34].

E. Clickjacking

Clickjacking is a malicious technique used by attackers when a user visits web page; the attacker forces him to click on random content which will redirect the web page to other web page and control the complete system access. This type of attack is known as clickjacking. This is a client-side security concern that affects the diversity of web browsers and platforms. The basic functionalities of these types of threats are to steal login user name passwords, spread malware, and promote online scams in social media [35].

F. Botnets

The botnet is a combination of bot and network. A bot for this situation is a device infected through a virus; it then becomes part of the network or network of infected devices well-controlled by a single attacker or attackers group [36]. The malicious botnet code usually inspects for vulnerable devices over the internet, instead of focusing on individuals, enterprises, or industries. The goal for making a botnet is to spread malware to as many connected devices as possible and to utilize resources of these devices for computerized tasks that are typically hidden from device users.

A summary of reported attack vectors is shown in Table I.

Table I. Summary of Attack Vectors

Attack vector	Description	Reference
Drive-by download	Malware silently installs itself into a system just accessing a vulnerable website.	28-29
Phishing Email	This type of attack is often used to steal user data, including login credentials.	30-31
Malvertising campaign	Malvertising is a short term for malicious advertising, and it uses legitimate online advertising services to spread malware.	32-33
Exploit kit	An exploit kit is a malicious toolkit cyber criminals use to attack vulnerabilities in systems so they can distribute malware or perform other malicious activities.	34
Clickjacking	This type of attack that tricks a user into clicking a webpage element which is invisible or disguised as another element.	35
Botnet	It is a combination of infected computers operated by a remote attacker, which is normally utilized for performing illegitimate activities.	36

6. Analysis and Detection of Encrypted Malware

As the encrypted traffic is deemed to grow, recognizing threats encapsulated within encrypted network traffic appears to be a challenging and cumbersome task. It is quite considerable to monitor encrypted traffic for threats and malware without compromising the user's secrecy. Since pattern matching is not so impressive in the case of TLS sessions, new methods need to be proposed that can detect malware communication precisely. However, TLS also provides an intricate set of observable data features, which can be utilized to detect and analyze malware communication, while protecting the secrecy of amiable applications of encryption also. This section reviews the malware analysis and detection in HTTPs traffic.

A. Features based Analysis of SSL/TLS Traffic

Petrvelan et al. [37] investigated the existing various encryption traffic protocol for the classification and analysis method. They discussed encryption protocol structure and working like IPSec (IP security), SSL /TLS, SSH (secure shell), and explored application- based protocol like Bit-Torrent and Skype. They revealed that most encryption protocols use an initial phase of information used for monitoring, inspecting, and encrypted traffic classification. Furthermore, the authors contributed an extensive survey of the behavior-based approach for encrypted traffic classification. A testing probing SSL security tool (PSST) was introduced in [38], and this tool was utilized to examine server security on the web, assessing more than 19000 servers. They evaluated various features on SSL/TLS server-side like version number, key exchange, and authentication, key size, symmetric encryption, and default choices. The PSST tool would be useful for checking the SSL/TLS server performance benchmarking. In this paper, the authors showed that some of the websites continue to support weak cipher suites and cryptographic keys. In future, they shall be working on the client-side as well so that they can evaluate more information on both sides (client and server) of the conversation. A complete study was conducted on the security concern of HTTPs interception by Zakirdurumeric et al. [39] and developed some heuristic rules based on TLS handshakes, security products, and TLS parameter applied to these set of heuristic rules to detect HTTPs interception. They deployed these heuristic rules on three different networks: (1) Firefox updated servers (2) E-commerce sites (3) Cloudflare content distribution network (CDN) and found that HTTPs interception causes other issues like privacy, security, etc. The authors envisage a new direction to the cyber security research community to find alternatives to HTTPs interception which does not violate the integrity of encryption.

For interception issues, Ngoc Huy Nguyen [40] in 2019 uncovered several SSL/TLS interception issues in organizations and discussed how security experts could manage all these issues. He used Wireshark, Tshark, and Zeek bro, tool [41] [42] to monitor client hello, client hello cipher suites, server hello certificates, and discussed their advantages and drawbacks. However, the researcher did not address other issues related to SSL/TLS handshake. Moreover, the rise of the TLS v 1.3 protocol on the internet provides extra security and privacy to the network security community. However, at the same time, it is not easy to inspect the TLS handshake process because some handshake features are encrypted in the TLS v 1.3 protocol.

To further address privacy and interception issues in HTTPs Wazen M. Shabir et al. [43] conducted the first examination of SNI (server name indication) deployment with a substantial arrangement of web servers that got to over HTTPs connections. The outcomes demonstrate that 92 percent of the HTTPs websites incorporated into the investigation are found with a forged SNI. To resolve this issue, they presented a new DNS based methodology to authenticate SNI with regard to HTTPs security monitoring that involves checking the relation between the legitimate destination server and the deserved value of SNI on the behalf of DNS service. In future work, they proposed to add this method in a firewall system for HTTPs.

The authors in [44] performed an experiment that inspected HTTPs traffic in a campus-wide LAN. They initially examined the parameters of SSL/TLS such as cipher suite, version number, and so on. They analyzed the connection between cipher suites lists and HTTPs user-agents. They allotted the similar User-Agents in the dictionary to the result of inspecting the SSL/TLS connections and discussed the dictionary's needed size and accuracy. This methodology was lightweight and avoided decrypting the traffic and improved the capabilities of network forensics by introducing the network-based identification of HTTPs clients while preserving the communication's privacy.

A group of indicators of malicious connections for SSL was proposed by Riccardo Bortolameotti et al. [45], which utilized unencrypted part of SSL. With the help of these indicators, authors found various malicious connections and vulnerable SSL connections to Man-in-the-Middle attacks (MITM). They also provided enough suggestions for the strength of their indicators to recognize malicious connections by verifying on the blacklists from professional web services. Furthermore, the prime concern of this work is to detect features that could indicate malicious behaviors. However, the authors did not use an ample dataset and more features to produce better results. In future work, they will try to implement an intrusion detection system based on this study. A novel method was proposed by Somnuk puangpronpitag et al. [46] to defend against SSL stripping attack. They discussed several techniques used by an attacker to hijack HTTPs session like SSL sniff and SSL strip. SSL sniff attack could be easily detected by digital certificate warning in the web browser. However, detection of SSL strip attack is difficult, so the authors proposed an ISAN-HTTPs-Enforcer method that could easily protect these types of attack, and checked proposed model compatibility in different web browsers and OS platforms.

B. Entropy based Techniques

Entropy based approach was too suggested by some of the researchers. Lyda and Hamrock [47] utilized entropy examination to distinguish encrypted and encoded malware; however, they just focused around offline executable files. The advantage of using entropy analysis is that it provides a convenient and fast technique for analyzing samples on the binary level, detects the executable file regions. The entropy is also used to detect uncertainty and randomness of the message [48].

Similarly, Dorfinger et al. [49] proposed an entropy-based method that actually works in real-time, as only the first packet of each network traffic flow is processed. The core classifier utilized payload entropy estimation, where entropy is used as a measure for uniformity, which is an indicator for encryption. In this paper, the entropy-based approach was used to categorize network traffic into enciphered and un-enciphered traffic. The enciphered traffic detected at an accuracy level of 94%.

C. Traffic Flow based Techniques

In order to detect the malware in encrypted traffic, J.M butler [50] described a brief overview of SSL protocol information and explored various methods used by an attacker to utilize encryption for a wrongful purpose. He introduced interceptor proxies between web clients and web servers, where encrypted traffic is decrypted and identified whether it contains malicious packets or not and then it is encrypted again to send it to the destination web server. Furthermore, Network visibility is a primary concern in this type of technique. To preserve privacy issues in network, Blake Anderson et al. [51] detected a threat in encrypted network traffic. Still, to do in such a way that sustains the integrity of the encryption. However, some features remained unencrypted like TLS handshake, DNS contextual flows related to the enciphered flow and the headers of HTTP associated with contextual flows. In this paper, they proposed a data Omnia approach, which means accurately classifying malicious TLS network flows and also giving useful information related to unencrypted metadata that could be useful to monitor and detect a threat in encrypted network traffic. The proposed model demonstrates high accuracy 99.97% at a 0.00% FDR. In a similar way, Blake Anderson et al. [52] conducted a comprehensive study of 18 malware families. They collected an enormous number of unique malware samples and tens of thousands of malicious TLS flows. They detected malware with the complex set of TLS features, without decrypting the TLS traffic while at the same time, it maintains the integrity of user security and privacy issues. Furthermore, the authors analyzed TLS parameters used by the malware from both the TLS client and TLS server side. The results have been shown to achieve a higher accuracy of 99% and a low false positive rate.

Tomas komarek et al. [61] addressed the issue of detecting end nodes infected with malware binaries that communicate via the HTTPs protocol. A set of features can be derived from HTTPs data without utilizing MITM (Man - In – The-Middle) method or DPI (Deep Packet Inspection). Authors proposed modeling based communication patterns whose prime objective was to observe through timing and transferred bytes. These patterns can be detected mainly in multiple connection requests which are issued by an individual end node. Authors focused on web proxy logs for utilizing information about the web traffic generated by end nodes in the monitored network. Furthermore, they conducted their experiment on real network data and demonstrated better efficacy results.

D. Machine Learning based Techniques

Conceptually similar work has also been carried out by [51] [52], which used TLS features to detect a threat in encrypted traffic. Several methods were proposed using a machine learning algorithm to detect and classify the encrypted traffic. Blake Anderson et al. [53] used various supervised machine learning algorithms to conduct their study and analysis of noisy labels and non-stationary data. They collected a lot of TLS encrypted flows over 12 months period through a commercial malware virtual box and two geographical distinct, big enterprise networks. From an algorithm and author's perspective, they found that the random forest ensemble method outperformed other methods; and showed that the feature selection policy had a more significant impact on performance. However, the authors heavily rely on human expertise to define the most relevant features. The proposed technique also shows the high accuracy at 99% with 0.01 FDR.

The authors introduced an intra-flow data approach in [54], which provides flow-based information to identify and monitor the network's threat. They have also released an open-source project joy tool [55], which shows all of the network flow monitoring techniques in this paper. They collected a huge amount of malicious flows from threat grid and benign flows from an enterprise networks DMZ. They analyzed various features related to network flow like a sequence of packet lengths and times, byte distribution and TLS handshake metadata and utilized these features for the creation of machine learning model and classify encrypted network flow. The result of using only flow based features was able to achieve 95.68% accuracy.

A feature analysis of encrypted traffic by Anish Singh et al. [56] classified it into two categories, namely, malicious encrypted traffic and benign encrypted traffic. Then, it compared the results from previous studies. The study also emphasized feature analysis based on machine learning models rather than human expertise to elaborate on the relevant features in encrypted traffic [53]. Furthermore, they trained and tested the models using three machine learning algorithms, namely, SVM, XGBoost, and random forest, and then performed feature analysis with the help of RFE (Recursive feature elimination) in each case. Out of these, XGBoost exhibited a little better performance than random forest, with the two achieving nearly 99% accuracy while SVM produced low accuracy comparatively. However, it would be better to experiment on a large dataset that could extract more useful and vital features.

The authors in [57] proposed a behavior testing method to identify HTTP and HTTPs network packets in a more detailed manner and applied machine learning methods to detect malware characteristics. They compared their results with a Cisco paper, to detect malware results quite similar to Cisco's paper but with different parameters. However, accuracy is not as effective as Cisco paper, but they managed to reduce the false-positive by 7%. The experimental results exhibit that precision and recall are more than 96% on average. In future work, authors will try to simulate results in real-time to detect malware, and further TLS metadata should be discovered in a more precise way.

A neural network based approach for malware detection in HTTPs network traffic is explored in [58]. In this approach, they developed an adaptable protocol that enables us to gather network flows of referred malicious and regular applications as training data and determine a malware detection technique in view of a neural embedding of domain names and a long short term memory (LSTM) network that processes network streams. Malware is detected in the context of the host address, timestamps, and data volume information of the computer network traffic. The authors concluded that LSTM based model outperform the random forest classifier. The experimental results show that the classifier achieves high precision 90% and recall 80% to detect malware in HTTPs traffic. Another similar study carried out by Lokoc et al. [59] designed a technique for the detection of malware in HTTPs traffic by K-NN classification. However, they focused on the problem of identifying malicious servers instead of understanding malicious traffic of several types. They extracted some statistical fingerprint features over HTTPs connection and used metric space search over high dimensional descriptors of network traffic to mitigate the false-positive rate. For future work, the authors are trying to implement data reduction techniques to enhance both the efficiency and effectiveness of the classification model.

The work presented in paper [60] is based on previously published works [59]. In this approach, they presented a statistical descriptor to identify the HTTPs communication pattern that would improve the detection of malicious activities by a machine learning algorithm. They also introduced map-reduce frameworks efficiently, building the descriptor, which extracted features from a considerable number of web traffic logs from corporate networks. The following features were used: bytes sent and received, duration, and inter-arrival time. In this paper, the proposed framework provides a scalable solution for malware detection in terms of 90% precision and 60% recall, thus providing an excellent way to deal with encrypted traffic analysis.

A framework proposed by J. Muehlstein et al. [62] to recognize the user's (O.S), browser, and web application traffic even though the network traffic is encrypted. They performed their experiment on the enormous amount of dataset more than 20,000 samples, explored familiar and new statistical traffic features, and time-series features in network traffic. The Authors applied a supervised machine learning algorithm for classification, which generated results with 96% accuracy. They extended their research to identify the operating system and browser behavior also in mobile devices.

The different approaches proposed by researchers for malware analysis and detection in HTTPs traffic are compared in terms of features, techniques, dataset, accuracy, evaluation metrics, and purpose of research and are summarized in Table II.

Table II. Comparison of Malware Detection Techniques in HTTPs Traffic

Authors, Year	Features	Techniques	Data set	Accuracy	Evaluation metrics	Purpose
Blake Anderson et al. [51]	TLS handshake metadata, DNS flows and HTTP headers.	Data Omnia approach	Cisco threat grid	99.97%	Accuracy, FDR,CV	Detect threats in encrypted traffic with high accuracy
Blake Anderson et al. [52]	TLS unencrypted header information	Logistic regression classifier	Enterprise Networks	99 %	Accuracy, Confusion matrix, FDR,CV	Detect threats in encrypted traffic without decryption
Blake Anderson et al. [53]	Statistical features of HTTPs connection	Machine learning	Enterprise Networks	99%	Accuracy, FDR,CV	Encrypted malware traffic classification
David Mcgrew et al. [54]	Sequence of packet lengths, byte distribution, TLS metadata.	Intraflow-flow approach.	Cisco threat grid and enterprise networks.	95.68%	Accuracy, FDR,CV	Detect threats in encrypted network traffic
Anish Singh et al. [56]	Statistical features of encrypted traffic	SVM XGBoost Random forest	CTU – 13 Dataset	92% 99% 98%	Accuracy, Specificity, Sensitivity, ROC,CV	Feature analysis of encrypted malicious traffic
Paul Culderon et al. [57]	HTTPs network packets	Weka framework	Georgia Institute of Technology	96%	Precision, Recall, CV	Conducted behavior analysis on HTTP and HTTPs packets.
Paul Prase et al. [58]	Network flow analysis and domain name features	LSTM model	Cisco Anyconnect secure mobility solution	Precision 90% Recall 80%	Precision-recall curve, ROC,CV	Detect threats in encrypted traffic with neural networks.
Jakub Lokoc et al. [59]	High-dimensional descriptors of network traffic	Metric space based approach	Cisco's cloud web security solution	Precision 88% Recall 50%	Precision, Recall, FP,CV	Reduced false positive rate.
Jan Kohout et al. [60]	Bytes sent, bytes received, duration, inter-arrival time	Map reduce framework	Corporate networks	Precision 90% Recall 60%	Precision-Recall curve, ROC,	Detect pattern for malware in HTTPs data.
Tomas komarek et al. [61]	Web proxy logs	Fingerprint based method	Cisco cognitive threat analytics	Precision 90% Recall 100%	Precision-Recall curve, ROC	Detect infected end-nodes of HTTPs network traffic.
Yehonatanzion et al. [62]	Statistical features of SSL and TCP connection	SVM based classifier	Ariel University	96.06%	Accuracy, Confusion matrix, CV	To classify encrypted network traffic.

7. EVALUATION METRICS

The performance of the malware detection in HTTPs traffic can be measured on the basis of various evaluation metrics. In this section, the evaluation metric used by various authors are reviewed. A list of detailed classifier evaluation metrics are defined below

A. Confusion matrix

It is a technique that is used to represent the performance of a classifier model [63]. There are 4 essential metrics involved in a confusion matrix is defined as below:

- **True Positives (TP):** It is the case when malicious traffic correctly classified.
- **False Positives (FP):** It is the case when benign traffic incorrectly classified as malicious traffic.
- **True Negatives (TN):** It is the case when network traffic is correctly classified as malware.
- **False Negatives (FN):** It is the case when network traffic is incorrectly classified as malware.

B. Accuracy

Accuracy is defined as correctly projected out of the entire test class [64]. It is also an integral factor for any classification model. The higher accuracy percentage is important for any implementation work. The accuracy is computed using the formulas given below.

$$\text{Accuracy}= \frac{TP+TN}{TP+FP+TN+FN} \qquad (1)$$

C. False Discovery Rate (FDR)

FDR is the rate at which true null hypotheses are rejected while conducting multiple comparisons [65]. It is also meant to decrease the fraction of false discoveries while sustaining a high true positive rate (TPR).

$$\text{FDR} = \frac{FP}{FP+TP} \qquad (2)$$

D. Sensitivity

It is the parameter that measures a model's capacity to predict the true positives of all available categories [66]. The sensitivity is computed by the formula as shown below.

$$\text{Sensitivity} = \frac{TP}{TP+FN} \qquad (3)$$

E. Specificity

It is the parameter that measures a model's capacity to predict the true negatives of all available categories [66].

$$\text{Specificity}=\frac{TN}{TN+FP} \qquad (4)$$

F. Precision

Precision is related to positive predictive value. It shows how good a model is in predicting the positive class [67].

$$\text{Precision}=\frac{TP}{TP+FP} \qquad (5)$$

G. Recall

Recall is also related to sensitivity. It points to the total percentage of relevant results out of correctly classified through an algorithm [67].

$$\text{Recall}=\frac{TP}{TP+FN} \qquad (6)$$

H. Precision-Recall curve

The precision-recall curve reviews the trade-off between the true positive rate (TPR) and the positive predictive value (PPV) of the prediction model utilizing several probability thresholds [68]. It is often used in information retrieval and useful for evaluating binary classification models that have an imbalance data for each class.

I. ROC curve

Roc curve shows the relationship between true positive rate (TPR) and false positive rate (FPR) of the prediction model utilizing several probability thresholds [68]. It is very useful for balanced data for each class.

J. K-Fold Cross-Validation

It is a statistical approach that evaluates and compares machine learning techniques by splitting data into train and test set [69]. The train set is used to learn or train a model and test set is used to validate the model. The formula to compute k-fold cross-validation accuracy measures as shown in eq. (7)

$$\text{CVA}=\sum_{i=1}^{k} A_i \qquad (7)$$

Where CVA stands for cross validation accuracy, k represents the number of folds and A also stands for accuracy measure for each fold [70].

A, Fig. 5 shows the accuracy factor for all research techniques that used in malware detection in HTTPs traffic. As shown, all of the accuracy factors higher than 90%. The highest accuracy percentage is 99% [51] [52] [53] and minimum accuracy percentage is 95% [54].

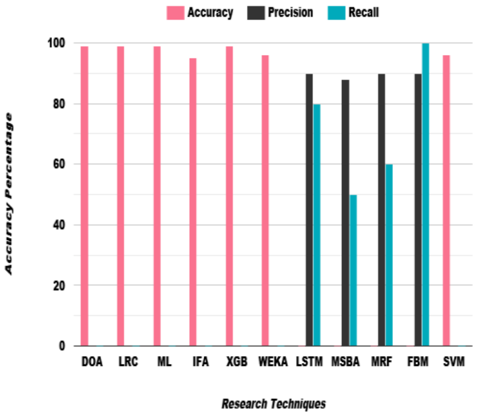

Figure 5. Accuracy Factor for Selected Techniques in Malware Detection

8. CONCLUSION AND FUTURE WORK

The current approaches for the investigation and detection of malware in HTTPs traffic are reviewed and explored in this paper. The amount of encrypted traffic is rapidly increasing day by day; therefore, detection of malware threats in encrypted traffic is a challenging and novel task. A malware threat can be realized by inducting several attack vectors into the network like drive-by-download and botnets etc. A conventional approach using a common interceptor proxy solution detects threats in encrypted traffic. However, at the same time, it violates the user’s privacy, and the communication process becomes computationally slow.

The comparative review of state of the art in malware detection in HTTPs traffic is performed. It reveals about the research strategies, features, data set, and evaluation metrics adopted by various researchers. A variety of techniques or approaches are used for the malware detection and analysis, with most researchers employing the machine learning based techniques. A few techniques have been reported having a low accuracy level or precision, while others have detected the malware with more accuracy. The selection of features is also significant in the process. Most of the techniques are using statistical features of the network traffic. The review also suggested that TLS metadata and DNS flow should be explored more deeply for improving the analysis and detection of malware in HTTPs traffic. Even if artificial intelligence techniques are effective for most of the obfuscation techniques, these cannot resist all highly advanced obfuscation and encoding techniques. The existing approaches can be extended by applying the deep learning based techniques for classification and detection of encrypted traffic.

REFERENCES

[1] Google Transparency Report. [Online]. Available: https://transparencyreport.google.com/https/overview. [Accessed: 21-Dec-2019].

[2] D. Akhawe, B. Amann, M. Vallentin, and R. Sommer,“Here’s mycert, so trust me, maybe? Understanding TLS errors on the web,” in Proceedings of the International Conference on World Wide Web, 2013.

[3] Turner, S. and T. Polk, “Prohibiting Secure Sockets Layer (SSL) Version 2.0”, RFC 6176 (Proposed Standard), March 2011.

[4] T. Dierks and E. Rescorla, “The transport layer security (TLS) protocol version 1.2,” RFC 5246, August 2008.

[5] Eastlake 3rd, D.,“Transport Layer Security (TLS) Extensions: Extension Definitions”.RFC 6066 (Proposed Standard), January 2011.

[6] Campbell-Kelly, Martin, “Historical Reflections: The Rise, Fall, and Resurrection of Software as a Service”, Communications of the ACM, vol.52 No.5, pp. 28–30, 2009.

[7] Cusumano, Michael,“Cloud computing and SaaS as new computing platforms”,Communications of the ACM,vol.53 No.4, pp.27–29, 2010.

[8] R. Dubin, A. Dvir, O. Pele, J. Muehlstein, Y. Zion, M. Bahumi,and I. Kirshenboim, "Analyzing https encrypted traffic to identify users operating system, browser and application" In IEEE Consumer Communications and Networking Conference, 2017.

[9] Wright, Charles V., Fabian Monrose, and Gerald M. Masson, “On Inferring Application Protocol Behaviors in Encrypted Network Traffic”, Journal of Machine Learning Research, vol.12 no.7, pp. 2745–2769, 2006.

[10] G. A. N. Mohamed and N. B. Ithnin, “Survey on Representation Techniques for Malware Detection System,” Am. J. Appl. Sci., vol.14, no. 11, pp. 1049–1069, 2017.

[11] Ihwail, Rami, Khairuddin Omar, and Khairul Akram Zainol Ariffin, "A Survey on Malware Analysis Techniques: Static, Dynamic, Hybrid and Memory Analysis." International Journal on Advanced Science, Engineering and Information Technology 8.4-2 pp.1662-1671, 2018.

[12] Y. Ye, T. Li, D. Adjeroh, and S. S. Iyengar, “A Survey on Malware Detection Using Data Mining Techniques,” ACM Comput. Survvol. vol.50, no. 3, pp: 1–40, 2017.

[13] M. Eskandari, Z. Khorshidpour, and S. Hashemi, “HDM-Analyser: A hybrid analysis approach based on data mining techniques for malware detection,” J. Comput. Virol. Hacking Tech., vol. 9, no. 2, pp.77–93, 2013.

[14] Panchenko, A., Lanze, F., Zinnen, A., Henze, M., Pennekamp, J.,Wehrle, K., Engel, T, "Website Fingerprinting at Internet Scale" In: Proceedings of the Network and Distributed System Security Symposium (NDSS), 2016.

[15] Wang, L., Dyer, K.P., Akella, A., Ristenpart, T., Shrimpton, "Seeing through Network-Protocol Obfuscation",In Proceedings of the ACM Conference on Computer and Communications Security (CCS). pp: 57-69, 2015.

[16] Zander, S., Nguyen, T., Armitage, G, "Automated Traffic Classification and Application Identification using Machine Learning", In The 30thIEEE Conference on Local Computer Networks. pp: 250–257, 2005.

[17] OmarYaacoubi,"The rise of encrypted malware." Journal on Network Security vol.2019 Elsevier issues.5 pp: 6-9, 2019.

[18] Sikorski,M., Honig, A "Practical Malware Analysis: The Handson Guide to Dissecting MaliciousSoftware".NoStarch Press, San Francisco, 2012.

[19] Kevin Fiscus, "Base64 Can Get You Pwned" The SANS Institute: Reading Room, 2011.

[20] Infosec Resources Malware Obfuscation, Encoding And Encryption. [online] Available at: <https://resources.infosecinstitute.com/category/certifications-training/malware-analysis-reverse-engineering/malware-obfuscation-encoding-encryption/#gref> [Accessed 05 May 2020].

[21] Roccia, T., 2020. "Malware Packers Use Tricks To Avoid Analysis, Detection"| Mcafee Blogs. [online] Available at: <https://www.mcafee.com/blogs/enterprise/malware-packers-use-tricks-avoid-analysis-detection/> [Accessed 05 May 2020].

[22] O'Kane, P., Sezer, S., McLaughlin, K "Obfuscation: the hidden malware". IEEE Secur. Privacy 9(5), 41–47 , 2011.

[23] Dark Reading. 2020. How Hackers Hide Their Malware: The Basics. [online] Available at: <https://www.darkreading.com/how-hackers-hide-their-malware-the-basics/a/d-id/1329722> [Accessed 08 May 2020].

[24] Musale, M., Austin, T. H., & Stamp, M. "Hunting for metamorphic JavaScript malware. Journal of Computer Virology and Hacking Techniques", 11(2), 89-102. 2015.

[25] Comar, P.M., Liu, L., Saha, S., Tan, P.-N., Nucci " A Combining supervised and unsupervised learning for zero-day malware detection". In: Proceedings of International Conference on Computer Communications, ser. INFOCOM. IEEE, pp. 2022–2030 2013.

[26] HexRays. Avaliable at: https://www.hex-rays.com/products/ida/[Accessed 10 May 2020].

[27] Helix Platform "Automatically Extracting Obfuscated Strings from Malware using the FireEye Labs Obfuscated String Solver (FLOSS)"[online] Availabe at: https://www.fireeye.com/blog/threatresearch/2016/06/automatically-extracting-obfuscated-strings.html [Accessed 10 May 2020].

[28] N. Provos, P. Mavrommatis, M. A. Rajab, and F. Monrose, "All your iframes pointto us" In 17th USENIX Security Symposium, 2008.

[29] "What is a 'Drive-By' Download?," McAfee Blogs, 15-Sep-2017. [Online].Available:https://www.mcafee.com/blogs/consumer/drive-by-download/. [Accessed: 18-Dec-2019].

[30] Parmar, Bimal. "Protecting against spear-phishing" Computer Fraud & Security. PP: 8–11, 2012.

[31] Courtesy of Computer Associates, "Types of Phishing Attacks," PCWorld, 12-Sep-2007. [Online]. Available: https://www.pcworld.com/article/135293/article.html. [Accessed: 05-Jan-2020].

[32] K. Z. N. W. E. KacyConnect, "Malvertising Campaign Delivers Millions of Bad Ads," Infosecurity Magazine, 30-Jul-2018. [Online]. Available: https://www.infosecurity-magazine.com/news/malvertising-campaign-delivers/. [Accessed: 29-Dec-2019].

[33] "Malvertising," CIS, 30-Sep-2019. [Online]. Available: https://www.cisecurity.org/blog/malvertising/. [Accessed: 1-Jan-2020].

[34] J. Segura and J. Segura, "Exploit kits: summer 2019 review," Malwarebytes Labs, 31-Jul-2019. [Online]. Available: https://blog.malwarebytes.com/threat-analysis/2019/07/exploit-kits-summer-2019-review/. [Accessed: 25-Dec-2019].

[35] "Protecting Your Users Against Clickjacking," Hacksplaining. [Online].Available:https://www.hacksplaining.com/prevention/click-jacking. [Accessed: 2-Jan-2020].

[36] M. Rouse, "What is a Botnet and How Does it Work?," SearchSecurity, 29-Oct-2019. [Online]. Available: https://searchsecurity.techtarget.com/definition/botnet. [Accessed: 15-Jan-2020].

[37] P. Velan, "A survey of methods for encrypted traffic classification and analysis", Networks, vol. 25, no. 5, pp. 355-374, 2015.

[38] H. K. Lee, T. Malkin, and E. Nahum, "Cryptographic strength of SSL/TLS servers: Current and recent practices" In IMC pp: 83-92, 2007.

[39] Zakirdurumeric, Zane Ma, "The Security Impact of HTTPS Interception" In Network and Distributed System Security Symposium (NDSS), 2017.

[40] Ngoc Huy Nguyen, "SSL/TLS Interception Challenge from the Shadow to the Light" The SANS Institute: Reading Room - Covert Channels, 2019.

[41] "Download," Wireshark · Go Deep. [Online]. Available: https://www.wireshark.org/. [Accessed: 12-Dec-2020].

[42] "The Zeek Network Security Monitor," Zeek. [Online]. Available: https://zeek.org/. [Accessed: 12-Dec-2020].

[43] W.M. Shbair, T. Cholez, J. Francois, I. Chrisment," Improving SNI-based HTTPs security monitoring", IEEE International Conference on Distributed Computing Systems Workshops, pp.72-77, 2016.

[44] M. Husák, M. Cermá, T. Jirsí, et al., "HTTPS traffic analysis and client identification using passive SSL/TLS fingerprinting,"Eurasip Journal on Information Security, pp. 1-14, 2016.

[45] R. Bortolameotti, A. Peter, M. H. Events, and D. Bolzoni, "Indicators of malicious SSL connections," in Network and System Security. Springer, pp. 162-175, 2015.

[46] Puangpronpitag, S., & Sriwiboon, N. "Simple and lightweight HTTPS enforcement to protect against SSL striping attack" In 2012 Fourth International Conference on Computational Intelligence, Communication Systems and Networks pp: 229-234, 2012.

[47] Lyda, r, Hamrock, j,"Using Entropy Analysis to Find Encrypted and Packed Malware"IEEE Security & Privacy vol.5 no.2 pp: 40-45, 2007.

[48] C.E. Shannon, W. Weaver, "The Mathematical Theory of Communication", 1963.

[49] Peter Dorfinger, "Real Time Detection of Encrypted Traffic Based On Entropy Estimation", Proceedings of the Third International Conference on Traffic Monitoring and Analysis, Austria pp: 164-171, 2011.

[50] J. Michael Butler, "Finding hidden threats by decrypting SSL" The SANS Institute: Reading Room, 2013.

[51] Anderson, B., McGrew,D, "Identifying Encrypted Malware Traffic with Contextual Flow Data". In ACM Workshop on Artificial Intelligence and Security, pp: 35-46, 2016.

[52] B. Anderson, S.paul and D.Mcgrew, "Deciphering Malware's Use of TLS (without decryption)" Arxiv Preprint ar Xiv: 1607.01639, 2016.

[53] Anderson, B., & McGrew, D. "Machine learning for encrypted malware traffic classification: accounting for noisy labels and non-stationarity". In Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining pp. 1723-1732, 2017.

[54] McGrew, D., & Anderson, B. "Enhanced telemetry for encrypted threat analytics". In 2016 IEEE 24th International Conference on Network Protocols (ICNP) pp. 1-6, 2016.

[55] D McGrew and B. Anderson. "Joy Software" Available at https://github.com/davidmcgrew/joy, 2016.

[56] A. S. Shekhawat, F. Di Troia, and M. Stamp, "Feature Analysis of Encrypted Malicious Traffic," Expert Systems with Applications Volume 125, Pages 130-141, 2019.

[57] Paul Calderon, "Malware Detection Based on HTTPS Characteristics via Machine Learning" In Proceedings of the 4th International Conference on Information Systems Security and Privacy pp. 410-417, 2018.

[58] Paul Prase, Lukas Machlica, “Malware Detection by Analyzing Encrypted Network Traffic with Neural Networks” LNCS, vol.10535, Springer pp. 73-88, 2017.

[59] J. Lokoc, J. Kohout, P. Cech, T. Skopal, and T. Pevny, “k NN Classification of Malware in HTTPS Traffic Using the Metric Space Approach” LNCS, vol. 9650, Springer pp.131–145, 2016.

[60] Jan Kohout, "Learning Communication Patterns for Malware Discovery in HTTPs data" Expert Systems with Applications, vol 101. Elsevier, pp. 129-142, 2018.

[61] Tomas Komarek and Petr Somol "End-node Fingerprinting for Malware Detection on HTTPS Data" In Proceedings of ARES ’17, Reggio Calabria, Italy pp. 1-7, 2017.

[62] J. Muehlstein, Y. Zion, M. Bahumi, I. Kirshenboim, R. Dubin, A. Dvir,and O. Pele, “Analyzing HTTPS Encrypted Traffic to Identify User Operating System, Browser and Application,” arXiv preprint pp.1-6, 2016.

[63] Brownlee, J., "What Is A Confusion Matrix In Machine Learning. [online] Machine Learning Mastery". Available at: <https://machinelearningmastery.com/confusion-matrix-machine-learning/> [Accessed 14 May 2020].

[64] T. Fawcett, “An Introduction to ROC Analysis,” Pattern Recognit. Lett., vol. 27, no. 8, pp. 861–874, 2006.

[65] Zhang, W., Kamath, G., & Cummings, R. . "PAPRIKA: Private Online False Discovery Rate Control". arXiv preprint arXiv:2002.12321, 2020.

[66] Medium. "Evaluating Categorical Models II: Sensitivity And Specificity". [online] Available at: <https://towardsdatascience.com/evaluating-categorical-models-ii-sensitivity-and-specificity-e181e573cff8> [Accessed 14 May 2020]

[67] Joshi, M., & Hadi, T. H. "A review of network traffic analysis and prediction techniques". arXiv preprint arXiv:1507.05722 ,2015.

[68] Brownlee, J., "How To Use ROC Curves And Precision-Recall Curves For Classification In Python"[online] Machine Learning Mastery. Available at: <https://machinelearningmastery.com/roc-curves-and-precision-recall-curves-for-classification-in-python/> [Accessed 14 May 2020].

[69] Wieland, M., & Pittore, M."Performance evaluation of machine learning algorithms for urban pattern recognition from multi-spectral satellite images". Remote Sensing, 6(4), 2912-2939,2014.

[70] D. Dursun, "Analysis of cancer data: a data mining approach," Expert Systems: The Journal of Knowledge Engineering, vol. 26, no. 1, pp. 100-112, February 2009.

Mr. Abhay Pratap Singh completed his B.Tech degree in Computer Science from Mangalayatan University, India in 2012, M.Tech from Manav Rachana International University in 2015. Now, He is pursuing Ph.D. from Gurukula Kangri Vishwavidyalaya Haridwar, India. He has also published various research papers in conference proceedings and journals. His research areas include Network Security, Malware Analysis, and Machine Learning.

Dr. Mahendra Singh is currently working in the Computer Science department of Gurukula Kangri Vishwavidyalaya, Haridwar, India as Asst Professor. He has diverse and vast experience of more than 20 years in the field of teaching/system administration. He has got master degrees in Physics and Computer Science. He obtained his Ph.D. in Computer Science in 2007 in the field of interconnection networks. He has got published more than thirty research papers indifferent National/ International Journals and conference proceedings. He has authored three books. His areas of interest are parallel and distributed computing, AI, Networks and Green Computing. He is a life member of Computer Society of India (CSI) and Indian Science Congress.