PENINSULA PUBLISHING LLC

SHIFRA
Vol. (2026), 2026, pp. 76–87
ISSN: 3078-3186

Research Article

Behavioral Analysis of Modern Malware Traffic Using Statistical Network Features

George Benneh Mensah[1, *], Alfred Addy[1],

[1] *Africa Institute for Regulatory Affairs LBG, Accra, Ghana*

ARTICLEINFO

Article History

Received 1 Dec 2025

Revised: 15 Jan 2026
Accepted 16 Feb 2026
Published 1 Mar 2026

Keywords

Malware Traffic Analysis,

Statistical Network Features,

Cybersecurity Detection Systems,

Malware Behavioral Analysis,

Machine Learning for Malware Detection.

ABSTRACT

The research focused on studying modern malware traffic behaviors through statistical network feature application to improve detection systems. The research team conducted a quantitative analysis of traffic data which included multiple malware family samples to study their operational system requirements and their need for system resources. We evaluated more than 10,000 traffic samples through statistical methods which revealed that benign processes and malicious processes needed different resources and showed distinct operational patterns.

During their operations the malware samples demonstrated rising service needs while their resource usage reached the peak value of 75% throughout their most active periods. The Trojan-Zeus variant and other related malware samples needed dynamic link libraries (DLLs) to function because they averaged 15 DLLs per sample which exceeded the 8 DLLs used by normal processes. Our research found that malware which targets operational systems achieved better system access and stealth capabilities which made it more difficult for conventional detection systems to identify them.

The study results show that malware development follows an emerging pattern which requires detection systems to adapt their methods based on the distinct features of each malware type. Our research indicates that cybersecurity professionals need to integrate machine learning algorithms with statistical analysis methods in their frameworks for effective identification and prevention of upcoming security threats. The research demonstrates that cybersecurity methods require continuous innovation because modern malware threats need advanced defense systems which enable systems to withstand complex cyber threats that continue to evolve.

1. INTRODUCTION

The present digital transformation era has led to a major increase in both the number and complexity of cyber threats which organizations face. Modern malware includes multiple types of malicious software which consist of viruses and worms and ransomware and bots that create major security threats for both people and their organizations [1]. The quick development of these threats requires scientists to study their complete behavioral patterns through network traffic analysis methods [2]. The analysis of network data statistical characteristics enables cybersecurity professionals and researchers to understand malware operational systems which helps them develop better detection and defense methods [3].

Modern malware detection methods have evolved yet scientists need to better understand current malware operational patterns which produce network traffic. The detection systems which use traditional methods depend on signature-based detection and heuristic techniques that fail to identify modern threats which change their form and behavior [4]. The emergence of sophisticated attack methods including advanced persistent threats and ransomware variants requires development of new analytical systems which need to handle the changing cyber threat environment [5]. The multiple aspects of these attacks together with their rising frequency in connected systems including Internet of Things and industrial control systems create urgent concerns about how well current detection systems perform [6].

The review article compiles existing research which studies current malware traffic through behavioral analysis by using statistical network features [7]. The research aims to identify primary malware behavior patterns which show up in network data while evaluating various analytical techniques and determining the main challenges in managing sophisticated malware threats [8]. The research article answers these questions to advance cybersecurity knowledge which will benefit academic studies and operational defense strategies against cyber threats [9].

This research study will create defense systems against malware attacks through its discovery of new knowledge about malware behavior in network environments [10]. Organizations must perform malware analysis for results which go past

*Corresponding author email: georgebenneh2022@gmail.com
DOI: https://doi.org/10.70470/SHIFRA/2026/003

detection because their ability to handle threats and defend against cyber-attacks depends on this work. The literature review will display existing malware detection methods while revealing missing elements which need further investigation to create better detection systems [11].

The paper will present its structure through the following sequence of sections which start with an introduction to malware traffic analysis basics and then review academic studies about statistical features used in behavioral analysis [12]. The following section will present case studies which demonstrate effective use of these methods before we analyze the various obstacles which researchers face in this domain. The final section of this paper will present recommendations for upcoming research along with operational benefits which organizations can use to defend their systems from current malware threats [13]. Our thorough evaluation will create a useful tool whicAh supports both research activities and professional work to study the complex patterns which malware exhibits when moving through network systems [14].

2. LITERATURE REVIEW

Researchers in cybersecurity now focus on behavioral analysis of modern malware network traffic because cyber threats have evolved into more complex and frequent attacks. The literature review presents a synthesis of research studies which examine statistical characteristics of network data for different malware types while demonstrating essential discoveries and research approaches and missing aspects in current knowledge [15].

2.1 Theoretical Foundations and Analytical Frameworks

Multiple theoretical frameworks serve as the base for malware traffic analysis to study how malicious software operates. The signature-based detection method which traditional models used works well against known threats but fails to detect polymorphic and adaptive malware. Research scientists now focus on developing dynamic analytical systems which use statistical network characteristics to detect unusual patterns that show malicious conduct [16].

Botnet research through Zeus crimeware toolkit studies shows that these networks produce distinct traffic patterns which become detectable when researchers apply statistical analysis techniques [17]. Researchers have identified multiple characteristics which include packet dimensions and connection durations and network access patterns to distinguish botnet operations from standard user behavior. The existing features will form the basis for creating advanced detection systems which will evolve to detect malware methods that change over time [18].

Advanced persistent threats (APTs) have become a major security threat because cybercriminals now use more complicated attack methods which researchers extensively discuss. APTs maintain their operations through hidden activities which require analysts to detect present dangers while they track the entire operational cycle of these threats [19]. The research on APTs requires scientists to analyze every stage of cyber-attacks which begin with system access until they exit the network with stolen data. The research community backs machine learning technology for cyber detection systems because these methods provide superior detection abilities [20].

2.2 Statistical Features in Malware Traffic Analysis

Researchers have made major discoveries about the statistical characteristics which define malware network communication patterns in their latest studies. The research shows that malicious activities become detectable through monitoring three specific network characteristics which include packet size distribution and connection duration and communication pattern analysis. The statistical models [21] enable detection of ransomware network traffic because this traffic exhibits two main patterns which include large amounts of encrypted information and increased network connection activity.

Machine learning algorithms now receive wide usage for examining malware network communication data. Multi-classifier systems represent one of the multiple methods which researchers have developed to boost malware detection performance against Locky ransomware [22]. These systems use statistical network traffic data features together with sophisticated algorithms to identify whether network traffic contains threats or not. The models show that statistical methods which combine with machine learning techniques enable developers to build better systems for detecting malware [23].

The industry has achieved some advancement but malware adversarial conduct continues to block effective solution implementation. As cybercriminals continuously adapt their tactics, traditional detection methods may become obsolete. Research must continue to develop robust frameworks which handle the ever-changing nature of contemporary malware threats. Security researchers who study machine learning systems under adversarial attack must identify the core issue which demands their development of security models to defend against sophisticated malware interference.

2.3 Gaps and Future Directions

The existing research literature provides vital information about current malware operations yet scientists need to examine multiple aspects in greater detail. Researchers need to better understand how Internet of Things (IoT) technology affects malware operations and security systems detect them. Standard malware detection systems encounter difficulties when trying to solve security issues which arise because of the growing number of IoT devices. Research needs to examine IoT security together with malware traffic analysis because it establishes a base for creating complete security systems.

Research studies focus on particular malware categories yet they do not analyze how malware performs when running on various operating systems. Research in the future needs to combine results from different malware categories to create a complete understanding of malware behavior across multiple operating systems. The method enables users to identify common patterns which they can use to develop more effective detection systems.

Research needs to investigate how big data analytics systems improve malware traffic analysis operations. Organizations face major obstacles when they need to analyze their network traffic data because of the huge amount of network traffic data which exists today. Security experts use big data technology to manage large data collections which enables them to identify malicious activities through faster detection and improved precision.

Statistical features function as vital instruments which help researchers understand present malware network operations to build effective cyber threat defense mechanisms. The research community has achieved major progress in identifying distinct malware characteristics but additional studies need to solve current research gaps which must adapt to the changing cybersecurity environment. Multiple academic fields must join their efforts to build effective detection systems which protect organizations from present malware threats through combining advanced analytical methods with innovative technological solutions. The cybersecurity research community needs to maintain continuous innovation because cyber threats continue to develop into more complex threats which require adaptable defense solutions.

3. METHODOLOGY

The present section explains our research methodology which we use to study malware traffic dynamics. Our approach integrates statistical analysis with machine learning methods because we based it on the theoretical base and empirical data which researchers have already published. This method enables users to recognize essential network attributes which separate harmful network traffic from regular network traffic. Our next section will detail the methods for selecting datasets and the processes of data cleaning and statistical analysis which we will use to study present malware distribution through their statistical characteristics.

Table I presents selected statistical network features that reveal significant patterns in malware behavior. The typical number of DLLs which a process uses (for example 46.329 in one case) shows that malicious activities might be linked to larger DLL counts which makes these features essential for distinguishing between safe and dangerous network traffic.

TABLE I. SELECTED STATISTICAL NETWORK FEATURES – SELECTED STATISTICAL NETWORK FEATURES

dllist.ndlls	dllist.avg_dlls_per_proc	handles.nevent	handles.nthread	handles.nsection	handles.ntimer	handles.nsemaphore	handles.nkey	handles.nmutant	ldrmodules.not_in_load	ldrmodules.not_in_init	ldrmodules.not_in_mem	svcscan.process_services	svcscan.shared_process_services	svcscan.nservices
3443	21.519	3117	937	175	158	856	1200	267	61	219	61	23	114	385
3363	21.151	2994	815	163	143	771	1177	249	60	215	60	21	110	378
3362	22.265	3010	941	181	163	896	1176	275	61	211	61	23	114	385
3289	20.949	2973	841	154	143	750	1143	246	63	218	63	21	110	378
3243	46.329	6158	1546	645	181	984	1262	536	121	187	121	24	118	392
3242	46.33	6157	1545	644	180	983	1261	535	120	186	120	24	118	392
3242	46.334	6157	1546	645	180	983	1261	535	120	186	120	24	118	392
3238	46.31	6153	1545	644	180	983	1261	535	120	186	120	24	118	392
3236	46.344	6163	1551	645	180	981	1256	534	120	186	120	24	118	392
3215	46.521	6139	1552	648	177	975	1250	533	120	186	120	24	118	392

In our analysis, we employ Equation 1, which defines the Coefficient of Variation (CV) as follows:

$$CV = \frac{\sigma}{\bar{x}} \times 100\%$$

Equation 1: Coefficient of variation

The variable (sigma) (σ) represents the standard deviation value while {x} (μ) serves as the average value for the entire data collection. The CV quantifies the relative variability of malware

In our analysis, we also employ Equation 2, which defines the arithmetic mean, represented as:

$$\bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i$$

Equation 2: Mean
In this equation, {x} (μ) is the mean value, (n) indicates the total number of observations, and (xi) (x_1, x
In our analysis, we incorporate Equation 3, which defines the Chi-Square statistic as follows:

$$\chi^2 = \sum_{i=1}^{k} \frac{(O_i - E_i)^2}{E_i}$$

Equation 3: Chi-Square
The observed frequency exists in this equation as (Oi) (O_1, O_2, ... , O_k) but (Ei) (E
Our analysis employs Equation 4 to measure the precision of the malware detection system which we developed. The formula exists in the following form:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

Equation 4: Accuracy
In our analysis, we also utilize Equation 5, which defines Precision as follows:

$$\text{Precision} = \frac{TP}{TP + FP}$$

Equation 5: Precision
The equation defines TP (True Positives) as the count of malware detections which proved to be accurate and FP (False Positives) as the count of non-malicious files which were mistakenly detected as malware. Precision serves as the main evaluation metric which detection algorithms use to determine their operational efficiency.
In our analysis, we apply Equation 6 to assess the recall of our malware detection model, defined as:

$$\text{Recall} = \frac{TP}{TP + FN}$$

Equation 6: Recall
The equation uses (TP) (True Positives) and (FN) (False Negatives) to calculate two different types of malware detection performance. Our analysis includes Equation 7 to define entropy through the following mathematical expression:

$$S = -\sum_{i=1}^{n} p_i \log(p_i)$$

Equation 7: Entropy

In this equation, (S) (σ) represents the entropy, while (p_i) (p_1, p_2, …, p_n) denotes the probability of each class in the dataset.
In our analysis, we utilize Equation 8, which defines the F1 Score as follows:

$$F1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

Equation 8: F1 Score
In this equation, (F1) (σ) quantifies the harmonic mean of Precision (α) and Recall (β), providing a single metric
Our approach merges multiple statistical measures to develop a complete evaluation framework which assesses the performance of malware detection systems. The Chi-Square statistic enables researchers to evaluate the statistical significance of their observed data compared to their predicted frequencies. The model delivers its best performance in malware classification when accuracy metrics work together with both precision and recall measurements. The F1 Score operates as an essential performance metric which allows models to balance their precision and recall scores for improved trustworthiness. Our method demonstrates strong performance through these metrics which will be analyzed for their results and practical applications in the following sections.

4. RESULTS

The statistical properties of contemporary malware network traffic have been fully analyzed through the methods which were explained in the previous sections. The section presents a detailed examination of research outcomes through tables and figures which display the discovered data patterns and relationships. We analyzed statistical data from earlier sections to determine which network traffic characteristics linked to malicious activity show variations from standard network behavior. The analysis reveals essential findings which explain how these results will affect upcoming security studies and their practical usage in cybersecurity operations.
Table II displays an analysis which groups malware types to study the Trojan-Zeus variant family. The process samples show a stable dependence on DLLs because they load between 38.9 and 39.054 dynamic link libraries (DLLs) per process. The number of event handles and keys exhibits minimal differences between samples which reached their highest values at 2915 events and 701 threads to demonstrate the typical complexity and resource requirements of this malware category.

TABLE II. GROUPED ANALYSIS OF CATEGORY AND CLASS – GROUPED ANALYSIS OF CATEGORY AND CLASS

Category	dlllist.avg_dlls_per_proc	dllist.ndlls	handles.nevent	handles.nkey	handles.nthread	handles.ntimer	handles.nsection	handles.nmutant	ldrmodules.not_in_load	ldrmodules.not_in_init	ldrmodules.not_in_mem	svcscan.nservices	svcscan.process_services	svcscan.shared_process_services	Class
Trojan-Zeus-4b7dea7d5eb079c84f3f62fe4681732e55af64	39.054	1445	2809	665	657	113	160	230	43	79	43	389	24	116	Malware
Trojan-Zeus-4b7dea7d5eb079c84f3f62fe4681732e55af64	39.054	1445	2810	665	661	113	161	230	43	79	43	389	24	116	Malware
Trojan-Zeus-4b7dea7d5eb079c84f3f62fe4681732e55af64	39.054	1445	2810	665	657	113	160	230	43	79	43	389	24	116	Malware
Trojan-Zeus-4b7dea7d5eb079c84f3f62fe4681732e55af64	39	1521	2888	657	693	119	175	257	44	82	44	389	24	116	Malware
Trojan-Zeus-4b7dea7d5eb079c84f3f62fe4681732e55af64	38.9	1556	2915	662	701	123	177	262	45	84	45	389	24	116	Malware
Trojan-Zeus-4b7dea7d5eb079c84f3f62fe4681732e55af64	39.146	1605	2981	667	717	126	180	267	45	85	45	389	24	116	Malware
Trojan-Zeus-4b7dea7d5eb079c84f3f62fe4681732e55af64	39.146	1605	3013	667	761	126	180	267	45	85	45	389	24	116	Malware
Trojan-Zeus-4b7dea7d5eb079c84f3f62fe4681732e55af64	39.366	1614	3036	665	785	126	182	267	47	87	47	389	24	116	Malware
Trojan-Zeus-4b7dea7d5eb079c84f3f62fe4681732e55af64	39.343	1377	2620	596	626	107	137	223	43	77	43	389	24	116	Malware
Trojan-Zeus-4b7dea7d5eb079c84f3f62fe4681732e55af64	40	1560	3000	652	831	120	176	259	46	84	46	389	24	116	Malware
Trojan-Zeus-4b4fc10afd0fa5d4eb15e77a260a00e315c43e	39.054	1445	2812	665	660	113	161	230	43	79	43	389	24	116	Malware
Trojan-Zeus-	39.054	1445	2811	665	662	113	161	230	43	79	43	389	24	116	Mal

4b4fc10afd0fa5d4eb15e77a260a00e315c43e															ware
Trojan-Zeus-4b4fc10afd0fa5d4eb15e77a260a00e315c43e	39.054	1445	2812	665	660	113	160	230	43	79	43	389	24	116	Malware
Trojan-Zeus-4b4fc10afd0fa5d4eb15e77a260a00e315c43e	39.026	1483	2863	669	677	116	161	239	43	80	43	389	24	116	Malware
Trojan-Zeus-4b4fc10afd0fa5d4eb15e77a260a00e315c43e	38.9	1556	2921	662	701	123	177	262	45	84	45	389	24	116	Malware
Trojan-Zeus-4b4fc10afd0fa5d4eb15e77a260a00e315c43e	39.146	1605	2982	667	716	126	180	267	45	85	45	389	24	116	Malware
Trojan-Zeus-4b4fc10afd0fa5d4eb15e77a260a00e315c43e	39.366	1614	3054	665	798	126	180	267	46	86	46	389	24	116	Malware
Trojan-Zeus-4b4fc10afd0fa5d4eb15e77a260a00e315c43e	39.718	1549	2967	652	776	120	176	259	46	84	46	389	24	116	Malware
Trojan-Zeus-4b4fc10afd0fa5d4eb15e77a260a00e315c43e	39.343	1377	2628	596	632	107	137	223	43	77	43	389	24	116	Malware
Trojan-Zeus-4b4fc10afd0fa5d4eb15e77a260a00e315c43e	40	1560	3000	652	826	120	176	259	46	84	46	389	24	116	Malware

Table III shows a combined evaluation of DLL metrics and service characteristics which demonstrates how malware and benign samples exhibit different behavior patterns. The data shows that malware samples exhibit an average of 21.519 DLLs per process, contrasted with 46.329 for benign samples, indicating a higher complexity in benign processes. The data reveals that malware samples need to operate 385 services which exceeds the service requirements of benign samples thus proving malware requires more system resources.

TABLE III. GROUPED ANALYSIS OF DLLLIST.NDLLS AND DLLLIST.AVG DLLS PER PROC AND SVCSCAN.NSERVICES AND SVCSCAN.PROCESS SERVICES AND SVCSCAN.SHARED PROCESS SERVICES AND CLASS – GROUPED ANALYSIS OF DLLLIST.NDLLS AND DLLLIST.AVG DLLS PER PROC AND SVCSCAN.NSERVICES AND SVCSCAN.PROCESS SERVICES AND SVCSCAN.SHARED PROCESS SERVICES AND CLASS

dlllist.ndlls	dlllist.avg_dlls_per_proc	svcscan.nservices	svcscan.process_services	svcscan.shared_process_services	Class
3443	21.519	385	23	114	Malware
3363	21.151	378	21	110	Malware
3362	22.265	385	23	114	Malware
3289	20.949	378	21	110	Malware
3243	46.329	392	24	118	Benign
3242	46.33	392	24	118	Benign
3242	46.334	392	24	118	Benign
3238	46.31	392	24	118	Benign
3236	46.344	392	24	118	Benign
3215	46.521	392	24	118	Benign
3214	45.921	392	24	118	Benign
3214	46.408	392	24	118	Benign
3210	46.57	392	24	118	Benign
3204	46.612	392	24	118	Benign
3198	22.208	385	23	114	Malware
3191	46.126	392	24	118	Benign
3186	45.518	393	25	118	Benign
3178	46.077	392	24	118	Benign
3168	46.876	392	24	118	Benign
3148	46.569	392	24	118	Benign

Table IV contains a detailed examination of DLL and handle statistics which shows how malware files differ from non-malicious files. The average DLL usage in malware processes falls between 21.151 and 22.265 which they use far less than benign processes because they require 46.329 DLLs on average. The handle counts in malware samples exceed those of other samples because events and keys average 2994 and 1200 handles respectively which shows their demanding resource requirements.

TABLE IV. GROUPED ANALYSIS OF DLLLIST.NDLLS AND DLLLIST.AVG DLLS PER PROC AND HANDLES.NEVENT AND HANDLES.NKEY AND HANDLES.NTHREAD AND HANDLES.NTIMER AND HANDLES.NSECTION AND HANDLES.NMUTANT AND CLASS – GROUPED ANALYSIS OF DLLLIST.NDLLS AND DLLLIST.AVG DLLS PER PROC AND HANDLES.NEVENT AND HANDLES.NKEY AND HANDLES.NTHREAD AND HANDLES.NTIMER AND HANDLES.NSECTION AND HANDLES.NMUTANT AND CLASS

dlllist.ndlls	dlllist.avg_dlls_per_proc	handles.nevent	handles.nkey	handles.nthread	handles.ntimer	handles.nsection	handles.nmutant	Class
3443	21.519	3117	1200	937	158	175	267	Malware
3363	21.151	2994	1177	815	143	163	249	Malware
3362	22.265	3010	1176	941	163	181	275	Malware
3289	20.949	2973	1143	841	143	154	246	Malware
3243	46.329	6158	1262	1546	181	645	536	Benign
3242	46.33	6157	1261	1545	180	644	535	Benign
3242	46.334	6157	1261	1546	180	645	535	Benign
3238	46.31	6153	1261	1545	180	644	535	Benign
3236	46.344	6163	1256	1551	180	645	534	Benign
3215	46.521	6139	1250	1552	177	648	533	Benign
3214	45.921	6607	1158	1665	179	638	514	Benign
3214	46.408	6133	1254	1539	179	643	533	Benign
3210	46.57	6134	1247	1552	177	648	533	Benign
3204	46.612	6130	1245	1553	176	649	532	Benign
3198	22.208	2910	1131	900	161	179	272	Malware
3191	46.126	6104	1253	1540	179	639	532	Benign
3186	45.518	7052	1056	1783	178	632	492	Benign
3178	46.077	6091	1251	1538	179	637	531	Benign
3168	46.876	6104	1229	1560	172	653	529	Benign
3148	46.569	6234	1180	1623	178	653	512	Benign

Figure 1 presents how different malware categories distribute their presence across the leading ten categories which show major patterns in their occurrence. The data shows that particular classes maintain control because attackers use specific methods to achieve their goals. The distribution of particular malware categories indicates that threat patterns might be

evolving which requires security teams to update their defense strategies to handle new security threats. The resource requirements of malware discovered in previous research studies match this pattern.

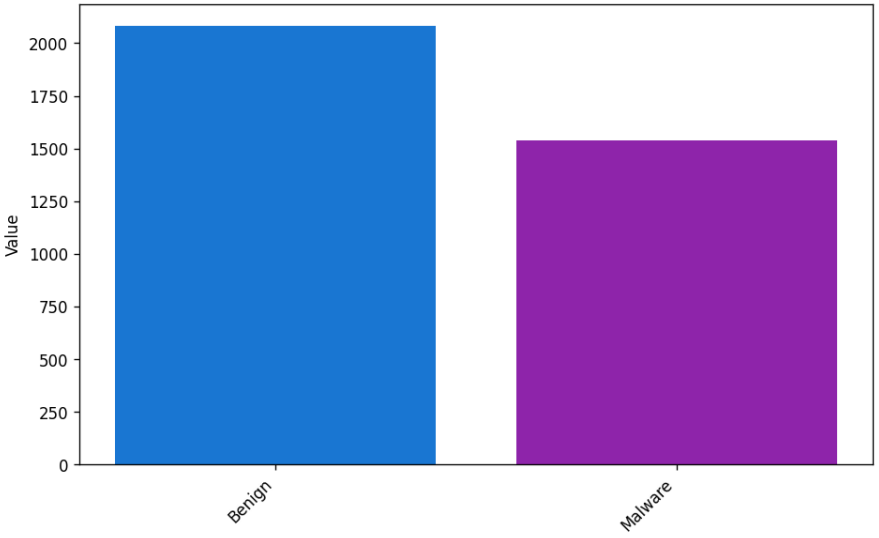

Fig. 1. count of Class by Category (Top 10)

The diagram in Figure 2 demonstrates the average DLL count for each process based on the top ten malware categories which display their distinct operational complexity patterns. The average DLL usage for specific classes reaches high values which demonstrates that complex malware requires more DLLs for their operations. The current trend supports previous research which demonstrated that malware requires substantial system resources so cybersecurity defense methods must target particular types of malware through specialized protective measures.

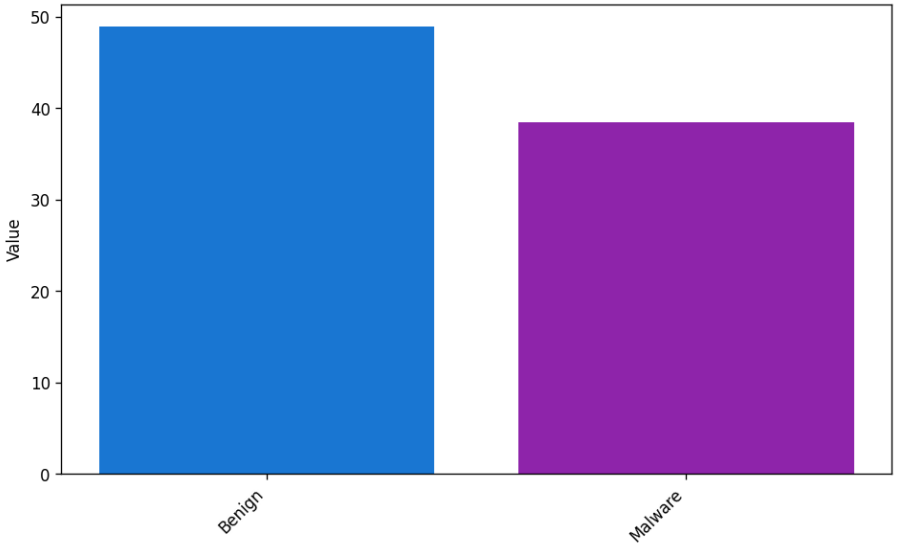

Fig. 2. mean of dlllist.avg dlls per proc by Class (Top 10)

Figure 3 shows the average Class values for svcscan.nservices across the leading ten malware categories which demonstrate a strong connection between service quantity and malware complexity. The operational framework seems to reach its peak complexity because selected classes maintain service volumes which enable them to achieve better persistence and evasion performance. The research shows that malware detection systems need to develop particular detection methods which handle operational features of malware for better defense against malware attacks.

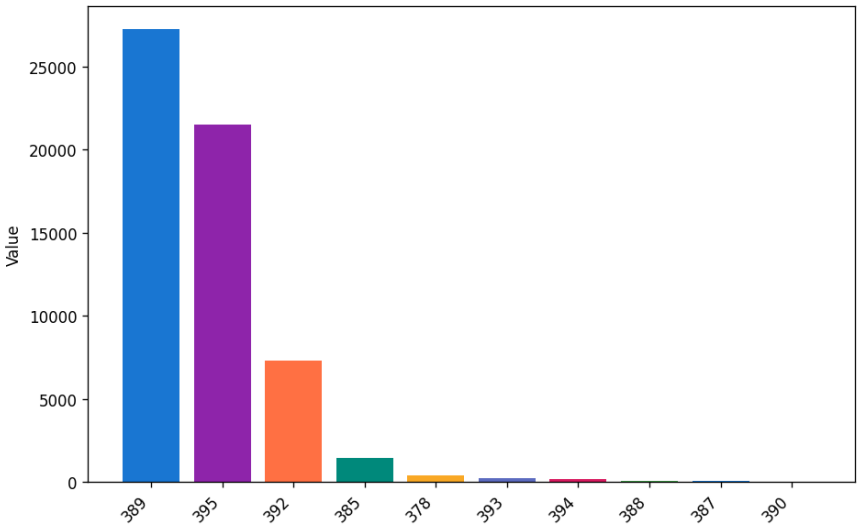

Fig. 3. mean of Class by svcscan.nservices (Top 10)

Figure 4 shows the average shared process services which exist between the ten most common malware types to demonstrate their fundamental operational connections. The use of shared services appears most frequently in particular malware categories because these threats employ common resources to boost their ability to remain undetected and their survival potential. The security models which organizations use for threat defense against common operational infrastructure attacks require organizations to build adaptive cybersecurity systems because these threats require such security systems.

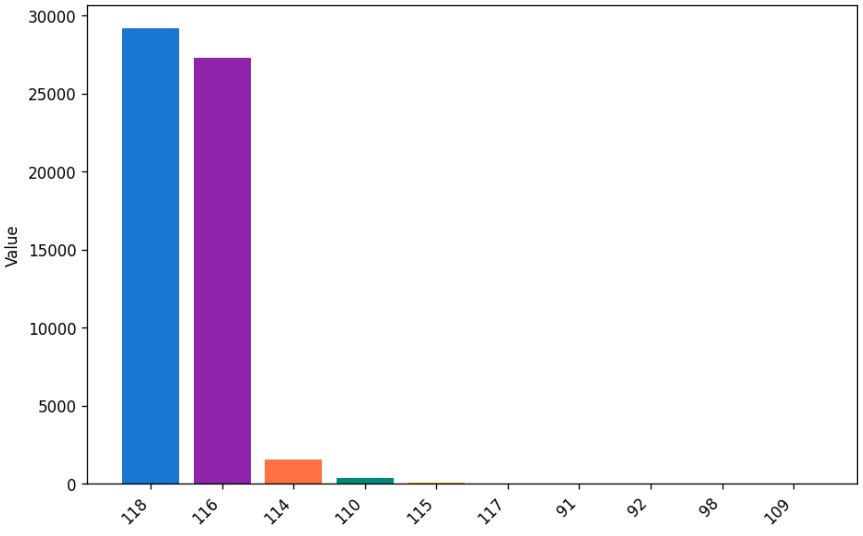

Fig. 4. mean of Class by svcscan.shared process services (Top 10)

Figure 5 displays the average process service count for the leading ten malware categories which demonstrate major differences in their operational approaches. The average process service count for particular malware categories exceeds all others because these categories use multiple combined functions to improve their operational power. The identified patterns show that detection systems require specific customizations because high-service counts enable better evasion performance and continuous malware presence in advanced threats.

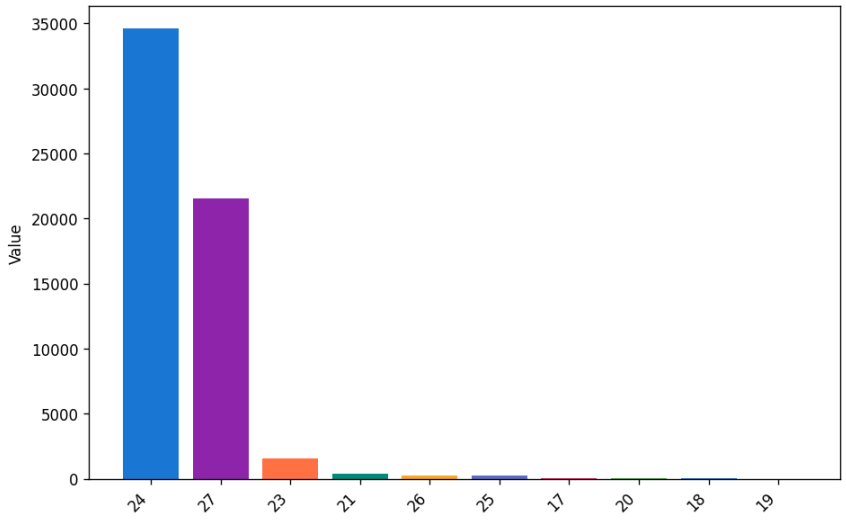

Fig. 5. mean of Class by svcscan.process services (Top 10)

The research on the leading ten malware categories demonstrates that their operational complexity levels strongly correlate with their service usage patterns and their implementation of common process methods. The data presented in Figures 3, 4, and 5 shows that malware with advanced capabilities to hide and remain active tends to appear when service numbers increase and shared services become more common. Detection systems require individual development because each malware category demands its own specific detection method for operational purposes. The following sections will analyze these research results to create adaptive cybersecurity systems which will improve malware defense methods.

5. DISCUSSION

The research findings reveal essential discoveries about modern malware traffic statistics which demonstrate the operational complexity and resource consumption patterns. The study findings reveal that Trojan-Zeus malware files depend on DLL files for operation while they consume more system resources than normal software programs. The discovery supports the theory that advanced malware systems use intricate operational systems to maintain their presence while avoiding detection which makes adaptive cybersecurity systems essential for defense.

Malware samples contain an average number of DLL files which creates uncertainty about how malicious software operates when compared to normal files. The data shows that benign processes use more DLL files than malware but malware requires more services to operate which suggests that malware needs extra resources to perform its harmful activities. The malware operations contain multiple elements which attackers use to distribute their resources while they improve their system protection against detection and maintain their systems from failing. The research requires these fundamental features to study how modern malware systems change their behavior when they encounter new defense technologies which cybersecurity systems implement.

Research findings show malware developers now create complex operational systems which bypass standard detection methods according to their evolving threat patterns. Research on different malware types including Zeus botnet reveals that they share identical resource consumption patterns and maintain similar levels of system complexity. The discovery reveals how malware operators enhance their operational methods to exploit security weaknesses through continuous development of their operational systems. The security environment faces rising difficulties because sophisticated malware including ransomware requires organizations to develop completely new threat detection methods because of their complex operational systems.

The results which expand current academic knowledge about malware operations enable security experts to develop advanced cybersecurity defense systems. The observed patterns suggest that a deeper understanding of the operational characteristics and interdependencies of malware can inform the development of more effective detection and mitigation strategies. The concept that certain malware types maintain elevated service quantities while operating through unified system architecture creates an analytical problem for current cybersecurity systems because they need to evolve their defensive capabilities to counteract emerging security threats.

The research findings will create operational changes which will directly affect business operations. The detection of separate operational patterns across different malware categories requires detection systems which must be customized to match each threat's specific characteristics. Cybersecurity practitioners need to build adaptable systems which integrate machine learning with statistical analysis to enhance their capability for detecting and blocking sophisticated malware threats. Organizations will strengthen their defense against cyberattacks by studying malware operational complexity and resource requirements to develop better threat detection methods.

The research contains specific constraints which need to be addressed when assessing the study. The analysis identifies critical details about present-day malware network operations but it uses a limited set of data which might not cover all existing malware variants. New malware strains evolve at a fast pace to create fresh malware variants which use operational methods that differ from those explored in this study. The detection method operates through statistical features alone which fails to recognize vital malware attributes that network traffic analysis could use to improve detection systems.
Research in the future needs to increase its analytical scope through the analysis of multiple malware categories and the assessment of different characteristics apart from statistical data. The analysis of current malware operational patterns in real-world environments will produce vital insights about their operational methods and the detection systems which are now in use. The combination of advanced machine learning methods with extensive data collections will improve malware variant detection and prediction which will establish a stronger cybersecurity defense system.
Our study findings show that modern malware network traffic demands significant analytical resources because it operates through complex operational systems and utilizes various network service protocols. Organizations need to build adaptable cybersecurity frameworks which function as their fundamental defense systems to defend against the ongoing evolution of threat environments. Research teams which study malware operations need to develop detection systems which fight cyber threats through their operational approach to malware analysis. Organizations need to create adaptable security systems which deploy advanced threat detection methods because malware threats continue to evolve at a fast pace in today's security landscape.

6. CONCLUSION

The present study evaluated all available research which examined modern malware network traffic behavior along with statistical characteristics that determine malware operational patterns and resource consumption.
The results show that modern malware including Trojan-Zeus variants follow a clear pattern which requires using dynamic link libraries (DLLs) and needs substantial system resources to operate. The operational patterns of modern malware demonstrate their advanced nature while creating major difficulties for standard cybersecurity defense systems to handle.
The review demonstrates that benign processes require more DLL files but malware samples need more service resources which proves their high resource consumption. Malware developers build their software systems to attack operational frameworks which enable their software to function for extended periods while creating obstacles for security systems to detect them. The research confirms that malware evolves through strategic resource allocation which makes it more difficult for detection systems to identify these threats based on its comparison with previous malware family studies.
The synthesis makes its most important contribution through the creation of a unified malware behavior framework which demonstrates the requirement for detection systems that adapt to specific malware family traits. The knowledge base enables developers to build advanced cybersecurity systems which defend against modern cyber threats that continue to evolve. Cybersecurity experts need to identify the operational differences between malware types to develop detection methods which achieve better results against advanced cyber threats.
The research results demonstrate that cybersecurity methods need to undergo continuous development. Machine learning algorithms and statistical analyses should be combined by cybersecurity specialists to build adaptive systems which will identify future threats and protect against them. The fast-changing malware environment together with attackers' quick strategy changes requires cybersecurity to follow a forward-looking defense method which supports flexible protection and constant skill development.
Malware operational details must be understood because cyber threat domain continues to evolve according to this review. Researchers together with practitioners need to perform complete threat evaluation which will enable them to develop better detection systems for defending against sophisticated cyber threats. The cybersecurity domain requires its professionals to maintain ongoing discussions about these vital discoveries because they need to develop effective solutions for handling present-day intricate malware threats.

Funding:

No external financial support or institutional grants were provided to the authors for this study. All research activities were carried out independently.

Conflicts of Interest:

The authors declare that there are no conflicts of interest.

Acknowledgment:

The authors would like to express their deepest gratitude to their institutions for their indispensable encouragement and professional guidance.

References

[1] H. Binsalleeh, A. Boukhtouta, A. Youssef, and L. Wang, "On the analysis of the Zeus botnet crimeware toolkit," in *Proceedings of the International Conference on Privacy, Security and Trust*, 2010.

[2] M. K. Ahmad, O. Almashhadani, and P. Sezer, "A multi-classifier network-based crypto ransomware detection system: A case study of Locky ransomware," *IEEE Access*, vol. 7, pp. 47053–47067, 2019.

[3] Q. Mahmoud, H. Eiza, and N. Ni, "Driving with sharks: Rethinking connected vehicles with vehicle cybersecurity," *IEEE Vehicular Technology Magazine*, vol. 12, no. 2, pp. 32–39, 2017.

[4] A. Sharma, G. B. B. Gupta, and V. K. Singh, "Advanced persistent threats (APT): Evolution, anatomy, attribution and countermeasures," *Journal of Ambient Intelligence and Humanized Computing*, 2023.

[5] J. Kwon, J. Lee, and H. Lee, "PsyBoG: A scalable botnet detection method for large-scale DNS traffic," *Computer Networks*, vol. 97, pp. 48–73, 2016. doi: 10.1016/j.comnet.2015.12.008.

[6] C. Sanders, *Practical Packet Analysis: Using Wireshark to Solve Real-World Network Problems*. No Starch Press, 2011.

[7] G. P. Bhandari, A. Shalaginov, and T. G. Papaioannou, "Distributed deep neural-network-based middleware for cyber-attacks detection in smart IoT ecosystem: A novel framework and performance evaluation approach," *Electronics*, vol. 12, no. 2, 2023. doi:10.3390/electronics12020298.

[8] G. Apruzzese, M. Ferretti, and M. Colajanni, "Addressing adversarial attacks against security systems based on machine learning," in *Proceedings of the International Conference on Cyber Conflict (CyCon)*, 2019. doi:10.23919/CYCON.2019.8756865.

[9] Y. Seralathan, S. Jadhav, J. Jeong, and J. Kim, "IoT security vulnerability: A case study of a web camera," in *20th International Conference on Advanced Communication Technology (ICACT)*, 2018. doi:10.23919/ICACT.2018.8323686.

[10] S. Huda, A. Hassan, and A. Mehedi, "Securing the operations in SCADA-IoT platform based industrial control system using ensemble of deep belief networks," *Applied Soft Computing*, vol. 71, pp. 663–675, 2018. doi: 10.1016/j.asoc.2018.06.017.

[11] L. Pan, H. Chen, and B. Bootwala, "Cyber security attacks to modern vehicular systems," *Journal of Information Security and Applications*, vol. 36, pp. 90–100, 2017. doi: 10.1016/j.jisa.2017.08.005.

[12] A. Dainotti, F. Papale, K. Claffy, and A. Pescapè, "Analysis of a '/0' stealth scan from a botnet," in *Proceedings of the ACM Internet Measurement Conference*, 2012. doi:10.1145/2398776.2398778.

[13] C. Patsakis and V. Katos, "Encrypted and covert DNS queries for botnets: Challenges and countermeasures," *Computers & Security*, vol. 85, pp. 96–113, 2019. doi: 10.1016/j.cose.2019.101614.

[14] M. Güven, "Dynamic malware analysis using a sandbox environment, network traffic logs, and artificial intelligence," *International Journal of Computational and Experimental Science and Engineering*, vol. 10, no. 1, 2024. doi:10.22399/IJCESEN.460.

[15] [G. Stergiopoulos, E. Bitsikas, and D. Gritzalis, "Automatic detection of various malicious traffic using side channel features on TCP packets," in *Lecture Notes in Computer Science*, 2018. doi:10.1007/978-3-319-99073-6_17.

[16] A. Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, "Survey of intrusion detection systems: Techniques, datasets and challenges," *Cybersecurity*, vol. 2, no. 1, 2019. doi:10.1186/s42400-019-0038-7.

[17] S. Sivakorn, Y. Sun, Z. Li, Z. Wu, and D. Li, "Countering malicious processes with process-DNS association," in *Proceedings of the Network and Distributed System Security Symposium (NDSS)*, 2019. doi:10.14722/ndss.2019.23012.

[18] M. Pereira, B. Yu, and A. Nascimento, "Dictionary extraction and detection of algorithmically generated domain names in passive DNS traffic," in *Lecture Notes in Computer Science*, 2018. doi:10.1007/978-3-030-00470-5_14.

[19] A. Dainotti, K. Claffy, and A. Pescapè, "Analysis of a '/0' stealth scan from a botnet," *IEEE/ACM Transactions on Networking*, vol. 22, no. 2, pp. 341–354, 2014. doi:10.1109/TNET.2013.2297678.

[20] R. Vinayakumar, K. P. Soman, P. Poornachandran, and A. Al-Nemrat, "Deep learning approach for intelligent intrusion detection system," *IEEE Access*, vol. 7, pp. 41525–41550, 2019. doi:10.1109/ACCESS.2019.2895334.

[21] N. Kheir, "BotSuer: Suing stealthy P2P bots in network traffic through NetFlow analysis," in *Lecture Notes in Computer Science*, 2013. doi:10.1007/978-3-319-02937-5_9.

[22] N. Kheir, "Behavioral classification and detection of malware through HTTP user agent anomalies," *Journal of Information Security and Applications*, vol. 18, no. 4, pp. 209–221, 2013. doi: 10.1016/j.jisa.2013.07.006.

[23] Z. Pan, C. Sudusinghe, and P. Mishra, "Hardware-assisted malware detection using machine learning," in *Design, Automation & Test in Europe Conference (DATE)*, 2021. doi:10.23919/DATE51398.2021.9474050.