electronics

MDPI

Article

LITNET-2020: An Annotated Real-World Network Flow Dataset for Network Intrusion Detection

Robertas Damasevicius *, Algimantas Venckauskas, Sarunas Grigaliunas, Jevgenijus Toldinas, Nerijus Morkevicius, Tautvydas Aleliunas and Paulius Smuikys

Faculty of Informatics, Kaunas University of Technology, 51386 Kaunas, Lithuania;
algimantas.venckauskas@ktu.lt (A.V.); sarunas.grigaliunas@ktu.lt (S.G.); eugenijus.toldinas@ktu.lt (J.T.);
nerijus.morkevicius@ktu.lt (N.M.); tautvydas.aleliunas@ktu.lt (T.A.); paulius.smuikys@ktu.lt (P.S.)
* Correspondence: robertas.damasevicius@ktu.lt

Received: 14 April 2020; Accepted: 11 May 2020; Published: 13 May 2020

Abstract: Network intrusion detection is one of the main problems in ensuring the security of modern computer networks, Wireless Sensor Networks (WSN), and the Internet-of-Things (IoT). In order to develop efficient network-intrusion-detection methods, realistic and up-to-date network flow datasets are required. Despite several recent efforts, there is still a lack of real-world network-based datasets which can capture modern network traffic cases and provide examples of many different types of network attacks and intrusions. To alleviate this need, we present LITNET-2020, a new annotated network benchmark dataset obtained from the real-world academic network. The dataset presents real-world examples of normal and under-attack network traffic. We describe and analyze 85 network flow features of the dataset and 12 attack types. We present the analysis of the dataset features by using statistical analysis and clustering methods. Our results show that the proposed feature set can be effectively used to identify different attack classes in the dataset. The presented network dataset is made freely available for research purposes.

Keywords: benchmark dataset; network intrusion detection; network attack; cyber security

1. Introduction

Network attacks are a set of network traffic events which are aimed at undermining the availability, authority, confidentiality, integrity, and other critical properties of networked computer systems [1]. Various types of cyber-attacks, such as IP spoofing [2,3] and Distributed Denial-of-Service (DDoS) flooding attacks [4], have been recognized as a serious security problem. With the increased scope, type and complexity of computer systems and communication networks, as well as with the emergence of new types of distributed computing technologies (such as the Internet-of-Things (IoT), Edge Computing, Fog Computing, etc.), new types of threats continue to arise against usual user requirements for privacy, security, and trust [5–8]. Despite numerous research studies in intrusion detection [9,10], there is still a large number of successful cyber-attacks registered each year, which affects the daily operation of businesses and governments, but also can cripple critical infrastructures [11], cloud-based IoT environments [12], cloud storage services [13], wireless sensor networks (WSN) [14], wireless body are networks (WBAN) in telemedicine systems [15], wireless ad-hoc networks [16], in-vehicle networks [17], software defined networks (SDN) [18,19], industrial IoT networks [20], cyber-physical systems [21], Internet of Drones (IoD) [22], Industry 4.0 smart factories [23], Internet of Medical Things (IoMT) such as implantable medical devices [24], IoT edge devices [25], vehicular ad hoc networks (VANET) [26], fifth-generation (5G) mobile networks [27], fog computing services [28], and user smartphones [29].

The increasing trend of cyber-attacks demonstrates that current intrusion detection methods still require improvement, and the development of new techniques is necessary for maintaining defense against cyber-attacks. Traditional security technologies and techniques often do note cope with emerging cybersecurity challenges in distributed, dynamic, heterogeneous, and wireless computing environments. As a result, there is a need to create new network security methods and systems to address the cybersecurity threats, as well as to collect, organize, and make openly available real-world heterogenous benchmark datasets [30] which would allow the comparison and validations of the capabilities of such systems in addressing the cybersecurity threats [31].

Network Intrusion Detection Systems (NIDS) are specifically designed to observe network traffic flows in order to identify any attacks and provide protection for the sensitive network infrastructure [32,33]. Each NIDS analyses network flows data, to check if there are any attacks. If there are any suspicious activities, the NIDS trigger an alert. However, usually a huge number of alerts is generated and deluges network security personnel [34]. The produced alerts often include irrelevant and redundant features, which result in higher resource consumption and lower attack-recognition accuracy. Moreover, uninformative features result in the reduction of attack-identification accuracy. Therefore, the dataset should contain only relevant features, allowing for efficient network-attack detection. The example of NIDS is a system for alarm correlation in Early Warning Systems (EWSs) [35] that uses port numbers, source and destination IP addresses, intrusion type, attack severity, and time stamp for alert generation.

The effectiveness of NIDS is evaluated based on their performance to recognize attacks, which requires a network dataset that provides examples of both normal and abnormal network traffic [36]. Old benchmark datasets such as KDDCup'99 [37] and NSL-KDD [38] have been widely used for evaluating the accuracy of network-attack recognition [39–42]. However, these datasets have become obsolete due to the rapid development of network technologies and the emergence of new cybersecurity threats and network attack types [33–45]. Many of the network flow benchmark datasets were created artificially, or the data were collected from a highly controlled environment, making them not representative of real-life network flows, while the methods trained on these datasets cannot cope with real-world network attacks [36–50]. Moreover, as the real-life network traffic grows enormously, manual data labeling of ever-increasing datasets becomes an infeasible task.

In supervised datasets, each observation in the dataset must have a label assigned to classify known attacks [51]. Unsupervised network-intrusion detection aims at detecting abnormal network node behavior, which can be attributed to a network intrusion or cyber-attack. For example, Casas et al. [52] employed a change detection method to identify the malicious network behavior. The flows are clustered, and the clusters are ranked by their abnormality; and the clusters exceeding the detection threshold are labeled as malicious. Another approach [53] uses entropy and Principal Component Analysis (PCA) to detect anomalies. Despite its successes, the unsupervised approach still requires large and up-to-date network traffic datasets reflecting real-life behavior of network nodes [54]. Umer et al. [55] employed a two-stage model for intrusion detection, in which a one-class Support Vector Machine (SVM) detects malicious flows, while a Self-Organizing Map (SOM) clusters of malicious flows into specific attack clusters. Unsupervised methods can potentially recognize unknown attacks with no prior knowledge, on which the supervised methods (which require datasets with data labeled by attack type) fail miserably [11]. To tackle these problems, Fadllulah et al. [56] extracted features for detecting attacks against encrypted protocols and generated normal-usage behavior profiles. The deviations from these normal profiles, which were identified by using the nonparametric cusum algorithm, were considered to be attacks. Zhang et al. [57] suggested using flow label propagation based on a Nearest Cluster based Classifier (NCC) to label network flows from an unlabeled dataset to convert the problem into supervised learning.

In order to develop new efficient network-intrusion-detection methods, the realistic and up-to-date network flow benchmark datasets are required [58]. Here, we present a new benchmark dataset collected from a real-world network and aimed at assessing the performance of NIDS in network-attack detection. Following the observation of Ring et al. [58], we aim to propose a dataset that is up-to-date, correctly annotated, publicly available, has real-world network traffic with multiple types of network attacks and examples of typical network behavior, and covers a considerable period of time. The dataset was created at KTU LITNET network from 06/03/2019 to 31/01/2020. The nfcap tool was used to capture network traffic. In order to create the traffic features, Nfsen, MeSequel, and Python script tools were used for feature generation. The dataset was annotated, to provide the examples of network-attack types. The key characteristics of the proposed dataset are a representation of real-world network traffic, which contains both normal network traffic and many examples of network attacks on real-world network infrastructure.

The remaining parts of the paper are structured as follows. Section 2 analyses the characteristics of known benchmark datasets and discusses their advantages and disadvantages. The environment of network data collection is described in Section 3. Section 4 presents a comprehensive comparison of the proposed dataset with other recently published network-intrusion datasets. Section 5 provides a detailed description of the files of the LITNET-2020 dataset. Finally, Section 6 presents conclusions and discusses further research plans.

2. Overview of Similar Datasets

The usability of any NIDS dataset reflects its power to provide information necessary to training the NIDS efficiently so that a high level of accuracy and reliability is achieved in detecting a diverse (as much as possible) set of network attacks. In this section, we discuss the main features of known network intrusion datasets (DDoS 2016 [43], UNSW-NB15 [59], CICIDS 2017 [60], UGR'16 [61], NSL-KDD [38], and CSE-CIC-IDS2018 [62]). Recently, several new network datasets have been proposed [63–66]. However, these have not yet been adopted by the research community as benchmark datasets. Therefore, hereinafter, for analysis, we use only datasets that have been widely adopted and used by the research community [58,67–69].

2.1. DDoS 2016

The dataset presents data collected in a controlled environment (using Network Simulator NS2), which has four malicious kinds of network attack: HTTP Flood, UDP flood, DDOS Using SQL injection (SIDDOS), and Smurf. The dataset has 27 features, 5 classes (4 attack classes and one normal traffic class) and 734,627 records.

2.2. UNSW-NB15

The UNSW-NB 15 dataset was generated by the IXIA PerfectStorm tool in a small network environment (only 45 unique IP addresses) over a short (31 h) period of time, and it includes a mix of real typical activities and artificial attack behaviors of the network traffic, resulting in 175,341 records for training and 82,332 records for testing. The IXIA tool simulated nine types of attacks. The dataset provides 49 features for analysis, which include basic features, content features (based on the content of packets), time features (based on time characteristics of packet flow), and additional generated features based on the statistical characteristics of connections.

2.3. CICIDS 2017

The dataset was made public by the Canadian Institute for Cybersecurity. The creation methodology used two types of usage profiles and multistage attacks, such as Heartbleed, and a variety of DoS and DDoS attacks. It has 80 network traffic features that are extracted by using the CICFlowMeter tool. User profiles were based on the abstract human behavior of 25 users working with the HTTP, HTTPS, FTP, SSH, and email protocols, aiming to generate the background traffic. The traffic was generated for a short (5 days) span of time.

2.4. UGR'16

This dataset was originated by the University of Granada (Spain) and is aimed for the assessment of cyclostationary NIDSs. The dataset was acquired from a tier-3 Internet Service Provider (ISP) over four months and has 16,900 million single directional flows. The real network traffic was mixed with synthetically generated malicious attack flows captured in a controlled network environment that somewhat decreases the quality of the dataset. It has 13 types of malware, including annotated botnet, SSH scan, and SPAM attacks, as well as background and normal network traffic, where background assumes that it is not known whether it contains a malicious traffic. The dataset was labeled by using the logs from the honeypot system.

2.5. NSL-KDD

NSL-KDD is an enhancement of the KDD dataset. In the KDD dataset, classification was biased toward more recurring records. However, in the NSL-KDD dataset, redundant items were removed, preventing the classifiers from achieving unreasonably high detection rates due to reoccurring records. The dataset includes 4 classes of attack: Denial of Service (DoS), Probe, User to Root (U2R), and Remote to Local (R2L). The training set has 4,898,431 records, and the testing set has 311,027 records.

2.6. CSE-CIC-IDS2018

The dataset covers six types of network attacks: Botnet, brute-force, Denial of Service (DoS), Distributed DoS (DDoS), infiltration, and web attacks. The dataset was generated based on the synthetic user profiles, which capture abstract representations of network events and behaviors. Fifty network nodes were used to organize an attack on the victim infrastructure with 420 computers and 30 servers. The dataset includes 84 network traffic features extracted from the network traffic, using the CICFlowMeter-V3 tool.

2.7. Summary

The comparison of the analyzed datasets by examples of attack types represented are summarized in Table 1. Here, Fuzzer aims to cause a network node suspended by transmitting to it the random data. Virus is a self-replicating malicious program that intrudes on the computer system without the knowledge of the user. Worm spreads through the network without the user's permission, while consuming network bandwidth resources. Trojan is a malicious program that causes the security problems in the network, while masquerading as a useful program. DoS aims to reject access to network nodes or resources for other users. Network Attack is an attempt to endanger network security from the data link layer to the application layer. Physical Attack attempts to cripple the physical units of computers or networks. Password Attack aims to obtain a password by login, and can be discovered by several login failures. Information Gathering Attack searches for known security holes by scanning or probing network nodes. User to Root (U2R) attack aims to take advantage of vulnerabilities of a network system in order to gain privileges as the super-user of the system. Remote to Local (R2L) attack dispatches packets to a remote computer system, without having a valid account on that system, aiming to obtain access either as a user or as a root. Probe attack scans the networks aiming to find valid IP addresses and to collect private data about the host, in order to start an attack on a selected set of systems and services.

The comparison of old reference (DARPA'98 [70] and KDDCup'99 [71]) datasets and more recent network intrusion datasets is presented in Table 2.

2.8. Conclusion of Dataset Analysis

New cyberthreats and types of attack continue to emerge. As a result, new realistic network datasets are needed to keep the development and benchmarking of network intrusion methods up-to-date. This has motivated us to collect network flow data from real-world network and to present it as an open benchmark dataset to be used freely for the research community in the cyber security domain.

Table 1. Types of attacks supported by the analyzed network-intrusion datasets.

Network Attack	Datasets					
	DDoS 2016 [43]	UNSW-NB15 [59]	CICIDS 2017 [60]	UGR'16 [61]	NSL-KDD [38]	CSE-CIC-IDS2018 [62]
Fuzzers	No	Yes	No	No	No	No
Generic	No	Yes	Yes	No	No	Yes
Virus	No	No	No	No	No	No
Worm	No	Yes	No	No	No	No
Trojan	No	Yes	No	No	No	No
DoS	Yes	Yes	Yes	Yes	Yes	Yes
DDoS	Yes	No	Yes	No	No	Yes
Network Attack	No	No	No	No	No	No
Physical Attack	No	No	No	No	No	No
Information Gathering Attack	No	Yes	No	Yes	No	No
User to Root (U2R)	No	Yes	No	No	Yes	No
Remote to Local (R2L)	No	No	No	No	Yes	No
Probe	Yes	No	No	Yes	Yes	No
Brute-force	No	No	Yes	No	No	Yes
Web	No	No	Yes	No	No	Yes
Infiltration	No	Yes	Yes	No	No	Yes
Botnet	No	No	Yes	Yes	No	Yes

Table 2. Comparison of reference and recent network intrusion datasets.

Data-Set	Reference Datasets		More-Recent Datasets			
	DARPA'98 [70]	KDD Cup'99 [71]	DDoS 2016 [43]	UNSW-NB15 [59]	CICIDS 2017 [60]	UGR'16 [61]
Year	1998	1999	2016	2018	2017	2016
Type of traffic	Synthetic network traffic	Synthetic network traffic	Randomized to obtain realistic results	Synthetic network traffic	B-Profile system	Real network traffic with realistic attacks
Raw binary Data	4 GB	n/a	n/a	100 GB of the raw traffic	n/a	14 GB
How it was col-lected	tcpdump	built from the DARPA'98 dataset	A network simulator (NS2)	tcpdump and IXIA traffic generator PerfectStorm	user behavior based on FTP, email, HTTP, HTTPS, and SSH protocols	netflow traces
Col-lection time	7 weeks	n/a	n/a	31 h	5 days	more than 4 months
No. of records	5 M records	4.9 M single connection vectors	734,627 records	2 Million	n/a	16,900 M
Label-ed attack types	n/a	DoS, User to Root (U2R), Remote to Local (R2L) and Probing Attack	DDoS attack (HTTP Flood, SIDDOS, UDP Flood, and Smurf)	Fuzzers, Backdoors, DoS, Exploits, Generic, ReconnaissAnce, Shellcode, Worms	Botnet, Brute Force SSH, DoS, DDoS, FTP, Infiltration, Heartbleed, Web Attack	DoS, Scan, Botnet (synthetic), IP in blacklist, UDP Scan, SSH Scan, SPAM, anomaly

3. Proposed Dataset

In this section, we describe the network environment used for the collection of network traffic data, provide the description of network attacks that are present in the dataset, provide the descriptive characteristics of the dataset, and discuss dataset preprocessing and availability.

3.1. Network Environment

Infrastructure consists of connecting nodes with operating equipment and communication lines connecting those exporters (CAPACITY, CYTI1, KTU UNIVERSITY 1, KTU UNIVERSITY 2, and FIREWALL). The LITNET NetFlow topology consists of two main parts (senders and collector). The NetFlow sender are Cisco (Cisco Systems Inc., San Jose, CA, USA) routers. Fortige (FG-1500D) high-performance-next generation firewalls analyze the data that has passed through it, process the data, and send it to one or more NetFlow server collectors. The NetFlow server (collector) is a server with appropriate software (nfcapd, nfdump, nfexpire, nfprofile, nfreplay, and nftrack. Version: 1.6.15), which is responsible for receiving, storing, and filtering data. We are used 4.9.0-11-amd64 #1 SMP Debian 4.9.189-3+deb9u2 x86_64 GNU/Linux operating system; 4 core Intel Xeon Processor (Skylake, IBRS) CPU processor; 10GB for system and 30T hard disks for collecting data; and 8 GB of RAM.

Each of the NetFlow Exporters (CAPACITY, CYTI1, KTU UNIVERSITY 1, KTU UNIVERSITY 2, and FIREWALL) continuously monitors the flows passing through it (this is a sequence of previous data packets in one direction from a specific sender to a specific recipient) and caches them when it receives new traffic. The main ring connects the five largest Lithuanian cities (see Figure 1), which are as follows: (CITY1) Kaunas–Vytautas Magnus University and Kaunas Technological University, which is administrator of Lithuanian Research and Education Network (LITNET) and maintenance and development connecting nodes and Vilnius University (CAPACITY); Vilnius Gediminas Technical University (CAPACITY); Klaipeda University (CITY2); Siauliai University (CITY3); and KTU Panevezys Faculty of Technologies and Business (CITY4). For efficiency reasons, the NetFlow sender only scans the first packet of the new stream, which saves the corresponding values, and then subsequent packets of the same stream are processed according to the same policy, thus reducing the load on the network device. Kaunas University of Technology (FIREWALL) has a high availability infrastructure and a perimeter Fortigate 1500D with FortiOS operating system, 80 Gbps network, IPS (intrusion prevention system) 13 Gbps, NGFW (next-generation firewall) 7 Gbps, and Threat Protection 5 Gbps bandwidth, firewall. CITY1 has an exit to broadband networks NORDUNET, and GEANT. CITY1 and CITY2 have a peering connection, and this is a process by which two Internet networks connect and exchange traffic. CITY1 and CITY2 can transfer data traffic directly between each other's LITNET users. Every other city (CITY2, CITY3, and CITY4) has end users. These are schools, municipalities, other organizations.

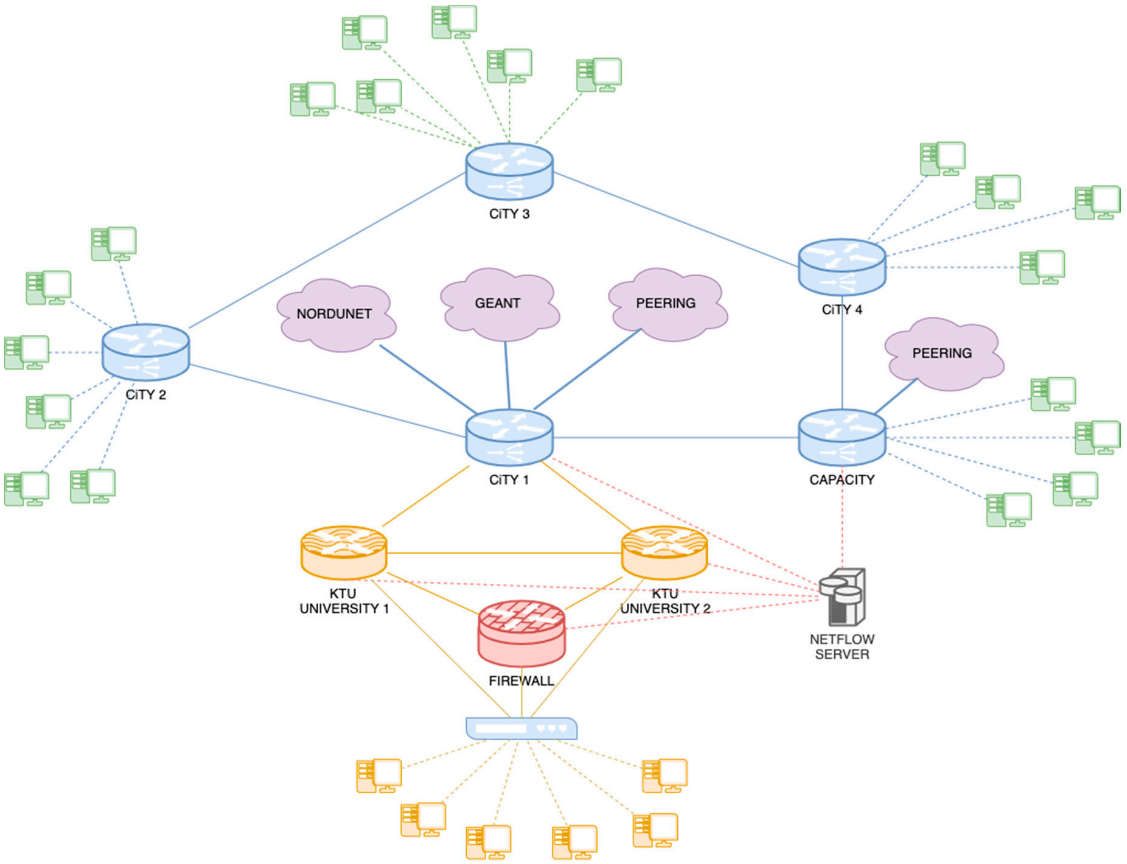


Figure 1. LITNET (Lithuanian Research and Education Network) network environment used for the collection of the dataset.

We used a 5 min time interval to make a nfcap (KTU UNIVERSITY1, KTU UNIVERSITY2, CITY1, and CAPACITY) file (nfcap is an application programming interface (API) for capturing network traffic with the format nfcapd.YYYYMMDDHHMM) and sent it to the NetFlow server to process information. In this time interval, we counted the number of packets, which satisfied the attack rules to distinguish the type of attack.

3.2. Description of Network Attacks

We describe the attack types in the proposed dataset as follows.

Smurf attack keeps sending the Internet Control Message Protocol (ICMP) broadcast requests to the network on behalf of the target node, aiming to flood the node with network traffic in order to slow down the targeted node.

An Internet Control Message Protocol (ICMP) flood attack is a DoS attack which aims to overwhelm a targeted network node with ICMP echo-requests (pings).

UDP-flood attack is a DoS attack using the User Datagram Protocol (UDP). A DNS Flood Attack (DNS Flooding) is an application-specific variant of a UDP flood, which is characterized by network packets sent to any IP address, using UDP protocol and port 53 as the target.

TCP SYN-flood attack is a Distributed DoS (DDoS) attack that misuses a part of the ordinary Transmission Control Protocol (TCP) three-way handshake to drain resources on the victim node and make it unresponsive. The attack packets packages have S flags but do not have the AFRPU flags.

HTTP-flood attack is a DDoS attack, which exploits seemingly legitimate Hyper Text Transfer Protocol (HTTP) GET or POST requests to assail a web server or application. In a complex Layer 7 attack, HTTP floods do not use ill-formed packets, spoofing, or reflection and need less bandwidth than other types of attacks, in order to disable the victim server or site. The attack packets are directed only to 80 port.

LAND attack is a Layer 4 DoS attack in which the malicious node sets the same source and destination data of a TCP segment. The attack packets have S flags and use the TCP protocol. An attacked node will hang due to the same packet being repeatedly processed by the TCP stack.

W32.Blaster Worm attack spreads by utilizing the Buffer Overrun Vulnerability of Microsoft Windows DCOM RPC Interface. The attacks are directed only to 135, 69 (TFTP), and 4444 (Kerberos) ports.

Code Red Worm attack aims to cause a buffer overflow problem on a target node, so that it begins to overwrite the adjacent memory. The packets are directed to source IP and only to 80 (no Secure Sockets Layer (SSL)) ports; this is how the HTTP GET method is applied.

Spam bot's attack dispatches spam messages or posts spam in social media platforms or forums. The packets are directed only to 25 (no SSL) port. The attack is characterized by the presence of an excessively large number of SMTP connections from one address.

Reaper Worm attack begins its last phase of scanning once the IP is passed to the exploit process. Reaper attack is directed at TCP ports 81, 82, 83, 84, 88, 1080, 3000, 3749, 8001, 8060, 8080, 8081, 8090, 8443, 8880, and 10,000. An attack is only recorded when the package contains the TCP stream and have not UDP or ICMP or ICMP6 protocols.

Port Scanning/Spread attack dispatches client requests to some server port addresses, aiming to discover an active port and taking of advantage of a known security hole. An abnormal number of connections from one host to one or more other hosts is as follows: several ports, one address; single port, multiple addresses.

Packet fragmentation attack is a kind of DoS attack, in which the attacker overloads a network by taking advantage of the datagram fragmentation.

3.3. Descriptive Characteristics

The traffic analysis is described for the cumulative flows while generating the dataset. The descriptive characteristics of the dataset are given in Table 3.

Table 3. Descriptive characteristics of dataset.

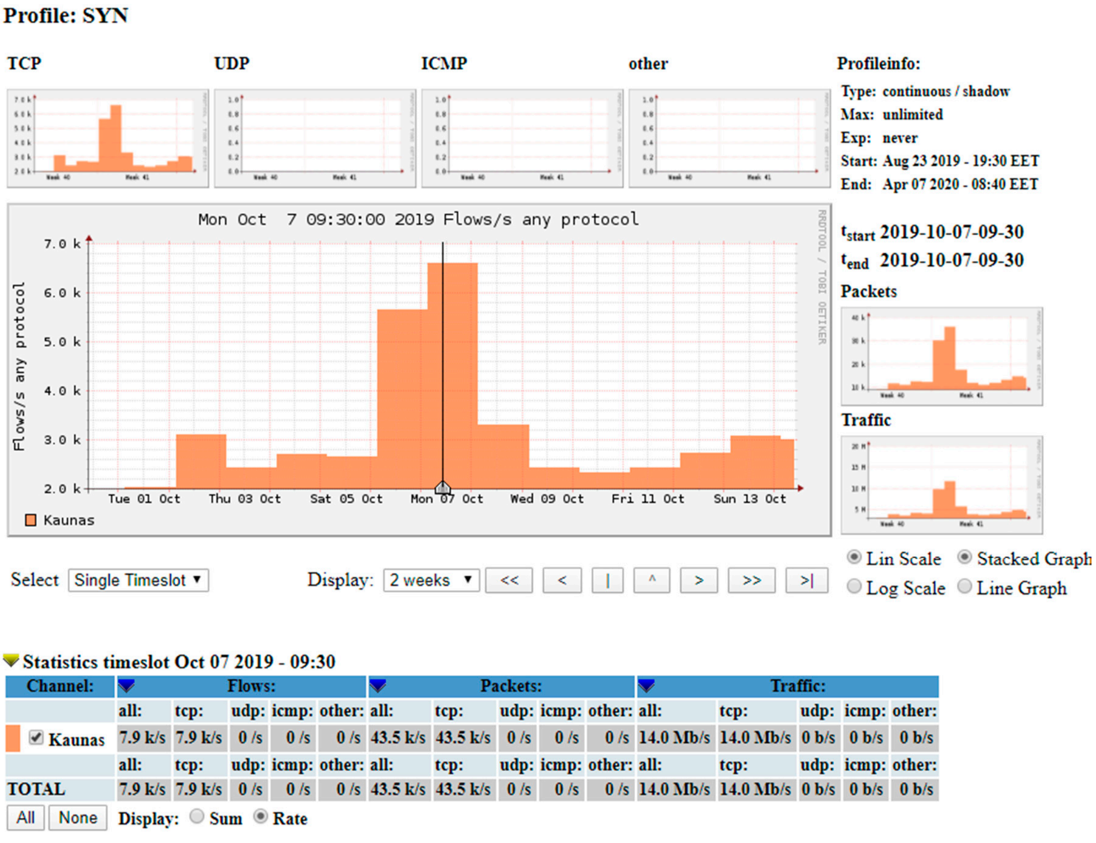
Characteristic		**Smurf**	**ICMP Flood**	**UDP Flood**	**SYN flood**	**HTTP Flood**	**LAND**	**W32.Blaster**	**Code Red**	**SPAM**	**Reaper Worm**	**Scan**	**Frag-mentation**
No of flows		59,479	11,628	93,583	3,725,838	22,959	52,417	24,291	1,255,702	747	1176	6232	477
Source bytes		15,471,966	2,217,910	18,506,962	2,195,053,120	15,138,945	30,257,200	11,636,600	433,310,270	1,247,220	276,960	1,893,620	116,883,130
Source packets		96,884	38,523	191,746	54,827,825	194,350	756,430	290,915	9,850,280	10,555	6924	36,410	149,595
Protocol type	TCP	0	0	0	3,725,838	22,959	52,417	24,291	1,255,702	727	1176	6323	0
	UDP	0	0	93,583	0	0	0	0	0	0	0	0	477
	IPv6	0	0	1	0	0	0	0	0	0	0	0	0
	ICMP	59,479	11,628	0	0	0	0	0	0	0	0	0	0
	Others	0	0	0	0	0	0	0	0	0	0	0	0
Unique IPs	Source	3865	1	8	7	1	3	1	873,211	1	1	1	5
	Destination	1	4900	5708	76,188	1	38,815	24,001	1	23	1153	6229	273
Unique ports	Source	1	1	46,531	4117	17,239	4	1	1	727	1128	6231	1
	Destination	5	3	1	230	1	6254	3	64,511	1	1	2	1

Table 4 represents the distribution of all data instances of the proposed dataset. All instances are categorized into ordinary data and attack data. The attack instances are further categorized into nine classes, according to the type of the network attack.

Table 4. Distribution of attacks in the dataset.

Type of Attack	Number of Flows	Number of Attacks (Flows)
Smurf	3,994,426	59,479
ICMP-flood	3,863,655	11,628
UDP-flood	606,814	93,583
TCP SYN-flood	14,608,678	3,725,838
HTTP-flood	3,963,168	22,959
LAND attack	3,569,838	52,417
Blaster Worm	2,858,573	24,291
Code Red Worm	5,082,952	1,255,702
Spam bot's detection	1,153,020	747
Reaper Worm	4,377,656	1176
Scanning/Spread	6687	6232
Packet fragmentation attack	1,244,866	477
TOTAL	45,330,333	5,328,934

A DoS attack with SYN packet can be explained in a simple way as the flow of illegal traffic to network resources from an IP address or the flow of IP addresses that results in a lack of network resources. The attackers disrupt the three-way click sequence by not responding to the SYN-ACK from the server, or they will constantly send a SYN packet from a non-existent IP, the server actually supports the queue set to which the SYN-ACK is sent because there will be no response from the clients, the queue will overflow, and the server will no longer be available. This is called a SYN Attack or Flood. The example of network traffic flows is shown in Figure 2.



Profile: SYN

Profileinfo:
Type: continuous / shadow
Max: unlimited
Exp: never
Start: Aug 23 2019 - 19:30 EET
End: Apr 07 2020 - 08:40 EET

t_{start} 2019-10-07-09-30
t_{end} 2019-10-07-09-30

Select: Single Timeslot
Display: 2 weeks
● Lin Scale ○ Log Scale
● Stacked Graph ○ Line Graph

Statistics timeslot Oct 07 2019 - 09:30

Channel:	Flows:					Packets:					Traffic:				
	all:	tcp:	udp:	icmp:	other:	all:	tcp:	udp:	icmp:	other:	all:	tcp:	udp:	icmp:	other:
☑ Kaunas	7.9 k/s	7.9 k/s	0 /s	0 /s	0 /s	43.5 k/s	43.5 k/s	0 /s	0 /s	0 /s	14.0 Mb/s	14.0 Mb/s	0 b/s	0 b/s	0 b/s
	all:	tcp:	udp:	icmp:	other:	all:	tcp:	udp:	icmp:	other:	all:	tcp:	udp:	icmp:	other:
TOTAL	7.9 k/s	7.9 k/s	0 /s	0 /s	0 /s	43.5 k/s	43.5 k/s	0 /s	0 /s	0 /s	14.0 Mb/s	14.0 Mb/s	0 b/s	0 b/s	0 b/s

Display: ○ Sum ● Rate

Figure 2. Example of network traffic flow.

Here, we can see that there is an obvious anomaly in sending SYN packets on the Kaunas (CITY1) channel. As you can see in the graph, this attack lasts two days (started from 2019-10-05 and ended on 2019-10-07). We also see that this attack occupies 14 Mb/s data traffic on the Kaunas (CITY1) channel. According to the presented real case, we can see that, on 2019-10-07 at 09:30, there was an attack peak of TCP SYN packets. This is evidenced by the huge number of packets in the data stream layout and the exceptional increase in data traffic on the graph.

For visualization of NetFlow, based on our proposed attack detection, the rules (see Figure 2; Profile: SYN) were developed for automatic notification of a possible cyber incident. For analysis of data stored in the collector (NetFlow server), we used flow-tools, nfstat, flowd, nfsen 1.3.6p1, php version-7.4.1, and Apache-2.4.25. MySQL database: version-10.1.41 (with 45 492 310 table records).

3.4. Dataset Availability and Preprocessing

For the collection of data, we use a methodology suggested in [61]. The network traffic data are captured in the nfcapd binary format files. The nfcapd files are collected in a single file per week for two capture periods. The mean size of files is about 1.35 GB (compressed). The nfcapd files have all NetFlow features, extended with 19 custom attack detection features which starts from 2019-03-06 first flow and 2020-01-31 last flow. The IP addresses of network nodes have been anonymized. Information from senders (NetFlow raw data files) to the collector is received in NetFlow v9 format (rfc3954). All values are transferred to the MySQL database (NetFlow database SQL server).

All dataset files can be freely downloaded from our website: https://dataset.litnet.lt.

The dataset formation is summarized in Figure 3. The data preprocessor selects 49 attributes that are specific to the NetFlow v9 (RFC 3954) protocol to form a dataset. The Data extender expands the generated dataset with additional fields of time, tcp flags, which are later used to identify attacks. The Extended dataset is supplemented by a set of 15 attributes. The generator creates additional 19 attributes for attack type recognition (see Table 5). The combinations of these and NetFlow attributes are used to detect attacks. We also added two additional fields to separate in the dataset, where the record is assigned to the attack and what specific type of attack, and where the normal network traffic is. Therefore, we have a total of 85 attributes.

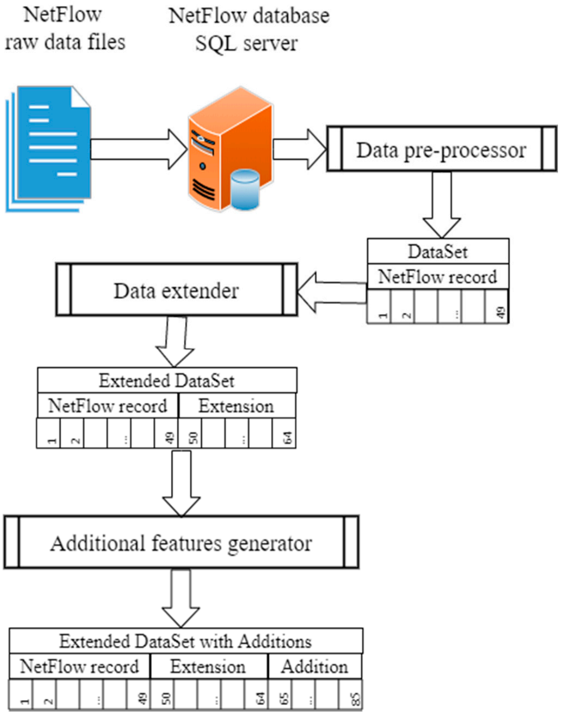


Figure 3. Processing of data and formation of the proposed dataset.

Table 5. Flow additional attributes for attack identification.

No.	Name of Features	Attribute	Description
1	icmp_dst_ip_b	icmp_smf	Flooding network broadcast with ICMP packets
2	icmp_src_ip	icmp_f	Flooding target with ICMP packets
3	udp_dst_p	udp_f	Ddos'ing with UDP traffic
4	tcp_f_s	tcp_syn_f	Flooding attack with SYN packets
5	tcp_f_n_a	tcp_syn_f	Flooding attack with SYN packets
6	tcp_f_n_f	tcp_syn_f	Flooding attack with SYN packets
7	tcp_f_n_r	tcp_syn_f	Flooding attack with SYN packets
8	tcp_f_n_p	tcp_syn_f	Flooding attack with SYN packets
9	tcp_f_n_u	tcp_syn_f	Flooding attack with SYN packets
10	tcp_dst_p	http_f	Ddos'ing with HTTP traffic
11	tcp_src_dst_f_s	tcp_land	Landing type of attack to any port with SYN packets
12	tcp_src_tftp	tcp_w32_w	Flooding TFTP service
13	tcp_src_kerb	tcp_w32_w	Flooding Kerberos service
14	tcp_src_rpc	tcp_w32_w	Flooding RPC service
15	tcp_dst_p_src	tcp_red_w	Uses a vulnerability in a HTTP server
16	smtp_dst	smtp_b	Flooding with SMTP connections from one host
17	udp_p_r_range	udp_reaper_w	Scans on UDP ports 80, 8080, 81, 88, 8081, 82, 83, 84, 1080, 3000, 3749, 8001, 8060, 8090, 8443, 8880, and 10,000.
18	p_range_dst	tcp_udp_win_p	Several ports, one address; single port, multiple addresses, ports NBT, Samba, MS-SQL-S, VNC, RDP, 2222
19	udp_src_p_0	udp_0	Ddos'ing with UDP fragmented traffic

3.5. Summary

The proposed dataset was collected in the real-world network, over an extended period of time (10 months), and contains real network attacks over the country-wide network infrastructure with servers in four geographically distributed locations (cities). As such, the proposed network flow dataset is more advantageous than some of its counterparts (such as UNSW-NB 15 dataset [59]), which generated the attacks artificially and thus do not contain realistic data.

4. Description and Statistical Analysis of Dataset Features

This section presents the description and analysis of dataset features. First, we formulate the requirements for dataset features. Then, we present the description of different classes of features. Next, we analyze the statistical distribution of the feature values and illustrate the results by figures. Finally, we summarize the results.

4.1. Requirements for A Dataset and Its Features

We formed the dataset by using the requirements for NIDS evaluation datasets outlined in [61] as follows. The dataset features should include network flow characteristics, such as IP addresses and port numbers, number of packets and bytes, flow duration, and flags. The dataset records should be correctly labeled as malicious or not, and in case of attack records, they should also include the type of attack. The dataset should cover several different periods of network activity, such as daytime/nighttime and weekdays/weekends.

4.2. Description of Features

In describing the features, we follow the description scheme suggested in [72] that considers the flow, basic, content, general purpose time slice, and connection features. These are summarized in Tables 5–7. The source_IP, target_IP, and time are noted as key for intrusion detection [73]. Source and destination ports, source and destination IP addresses, and time were mentioned as the most informative features for intrusion alerting [19]. Similar time-slice features based on the calculation of unique IP addresses within a time window were successfully used for network-attack detection before [74]. Additionally, we provide two network-attack attributes labeled by the network security experts, in Table 8.

Table 6. Additional generated attributes.

No.	Type of Attack	Action	Attribute	Description
1	Smurf	Flood	icmp_smf	Broadcast requests to the network on behalf of the victim computer
2	ICMP-flood	Flood	icmp_f	Large traffic of ICMP packets
3	UDP-flood	Flood	udp_f	Large traffic flow to DNS
4	TCP SYN-flood	Flood	tcp_syn_f	Large traffic of TCP traffic with SYN attack
5	HTTP-flood	Flood	http_f	High traffic with HTTP protocol
6	LAND attack	Attack	tcp_land	The IP address of the target is indicated in the header of such an IP packet as the destination and departure addresses, and any open port on the system under attack is indicated as the destination and departure ports
7	W32.Blaster Worm	Worm	tcp_w32_w	Large volume of traffic on Remote Procedure Call (RPC) port, for TFTP port, Kerberos authentication port
8	Code Red Worm	Worm	tcp_red_w	Uses a vulnerability in a web server
9	SAPM bots	Bots	smtp_b	The presence of an excessively large number of SMTP connections
10	Reaper Worm	Worm	udp_reaper_w	Reaper is a botnet that uses the HTTP-based exploits of known vulnerabilities in IoT
11	Scanning/Spread	Attack	tcp_udp_win_p	An abnormal number of connections from one host to one or more other hosts
12	Packet fragmentation	Attack	udp_0	Denial of service attacks are based on the use of many fragmented packets

Table 7. Time-slice-based connection features.

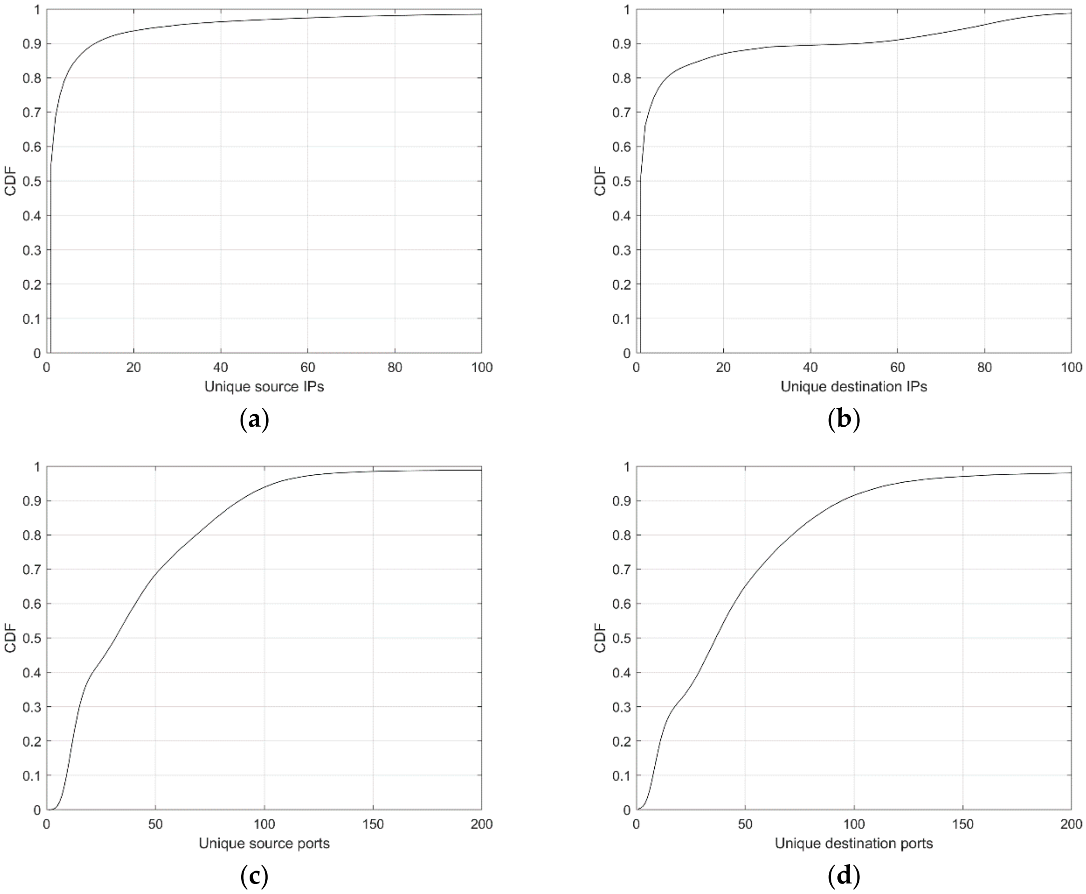
No.	Description
1	No. of unique source IP addresses in previous 10,000 connections
2	No. of unique destination IP addresses in previous 10,000 connections
3	No. of unique source ports in previous 10,000 connections
4	No. of unique destination ports in previous 10,000 connections
5	The largest count of connections from the same source IP address in previous 10,000 connections
6	The largest count of connections from the same destination IP address port in previous 10,000 connections
7	The largest count of connections from the same source port in previous 10,000 connections
8	The largest count of connections from the same destination port in previous 10,000 connections
9	The average count of connections from the same destination IP address and port in previous 10,000 connections
10	The average count of connections from the same source port in previous 10,000 connections
11	No. of connections with unique source IP–destination IP address pairs in previous 10,000 connections
12	The largest count of connections with the same pair of source IP–destination IP addresses in previous 10,000 connections

Table 8. Labeled attributes.

Name of Features	Description
attack_t	(attack type) nine types: Smurf, ICMP-flood, UDP-flood, TCP SYN-flood, HTTP-flood, LAND attack, W32.Blaster Worm, Code Red Worm, SAPM bots, Reaper Worm, Scanning/Spread, Packet fragmentation or none
attack_a	(attack action) 0 for typical (background) traffic and 1 for network attack

4.3. Analysis of Features

Following [75], we calculated the mean and standard deviation of all features. We studied the feature variance, using the cumulative distribution function (CDF), as was suggested in [76]. The results are shown in Figure 4.



(**a**) (**b**) (**c**) (**d**)

Figure 4. Feature value distributions observed in network flows as captured in the proposed dataset: (**a**) unique source IPs, (**b**) unique destination IPs, (**c**) unique source ports, and (**d**) unique destination ports.

These feature distributions are heavy tailed, and the smaller values make most cases. Moreover, 95% of source network nodes connect to fewer than 29 unique destination IPs, only 1% of source nodes may connect to more than 185 unique IPs, and only 0.1% connect to more than 2900 unique IPs (SYN attack subset of the dataset). Such outliers help in identifying the malicious behavior in the network. Feature value distributions also reveal possible correlations between features. For example, the Pearson correlation between input packets and input bytes is 0.92, when an attacker performs the SYN attack. Such high levels of correlation allow us to identify features containing duplicate information.

To analyze the dynamical changes in the network flows, we applied window slices and observed the change of unique source and destination IPs over time. Here, we used a window of 10,000 NetFlows moved with a step of 5000 NetFlows. The results are presented in Figure 5 for the IP addresses and in Figure 6 for the port connections. One can see sharp changes in the behavior of network nodes, which may be indicative of the network attack.

The distribution of data according to the protocol types is presented in Figure 7. The most common protocols in the dataset are TCP and UDP.

The temporal frequency of the source and destination ports in the NetFlow connections are presented in Figure 8. As a baseline, a reference frequency is given, if the distribution of connections to ports would be uniform. Note that connections from/to some ports are much more frequent, e.g., the most frequent source ports are 54,438 and 444, while the most frequent destination ports are 444 and 54.

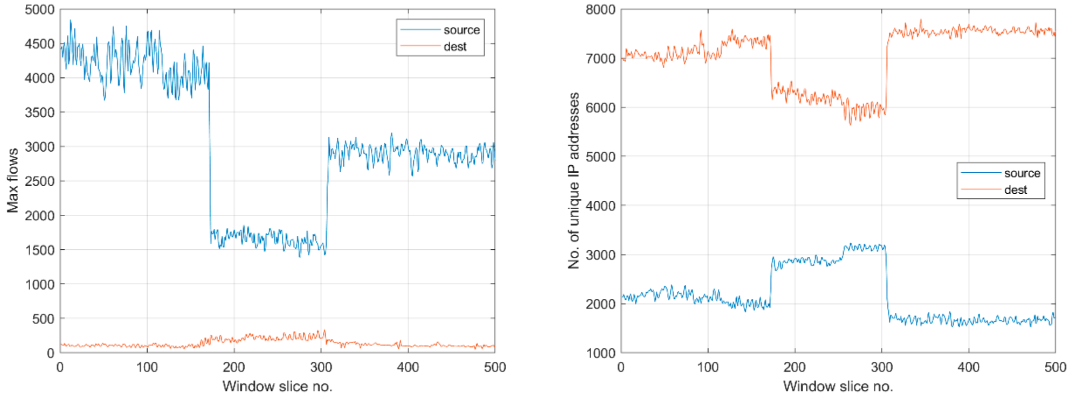


Figure 5. Maximum flows and number of unique IP addresses in NetFlow window slices.

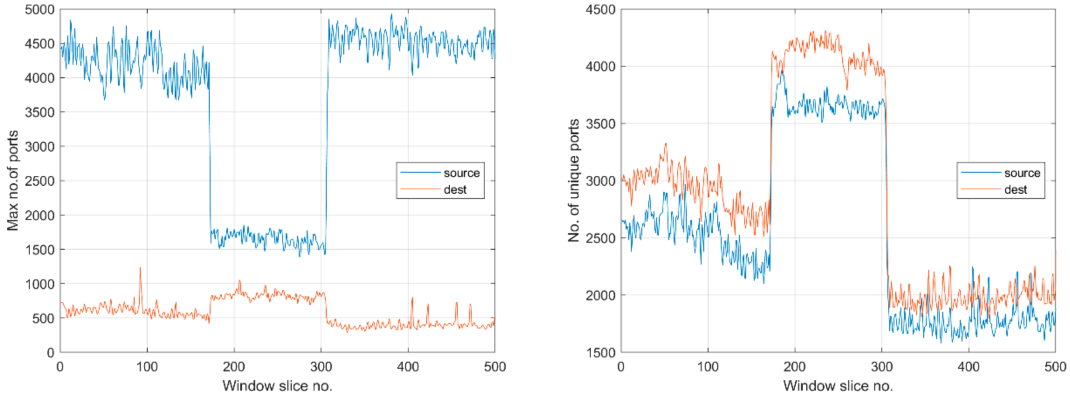


Figure 6. Maximum number of ports and the number of unique ports in NetFlow window slices.

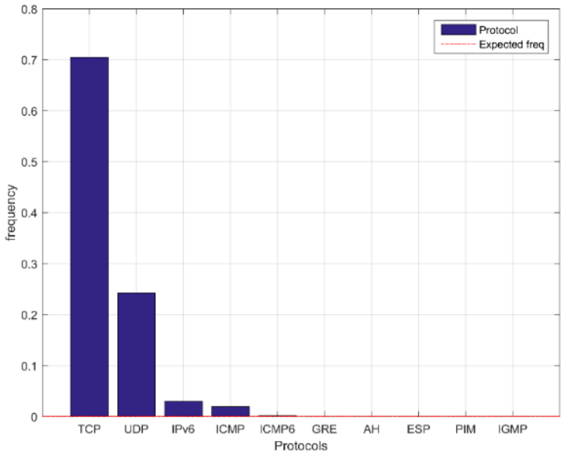


Figure 7. The distribution of data, according to the protocol types.

The statistical distribution of the connection flags in the TCP SYN-flood subset of the dataset is presented in Figure 9. It shows that most of the network connections had the S (TCP SYN) flag.

In case of the scan spread attack, the attacker scans for new IPs in the network. As a result, the number of unique destination IPs in a NetFlow window slice grows steadily during the attack (see Figure 10, left). Moreover, the attacker performs the port scan on the attack nodes, which can be seen from the distribution of port numbers in time (see Figure 10, right).

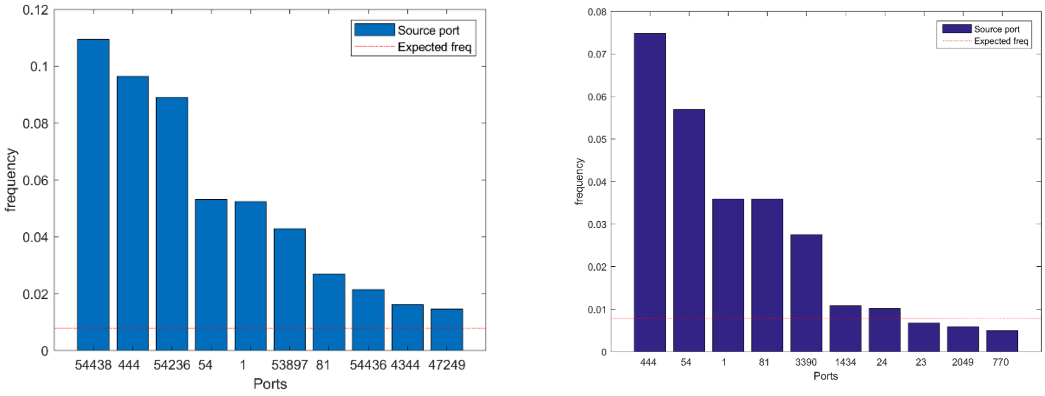


Figure 8. The frequency of the source and destination ports in the NetFlow connections.

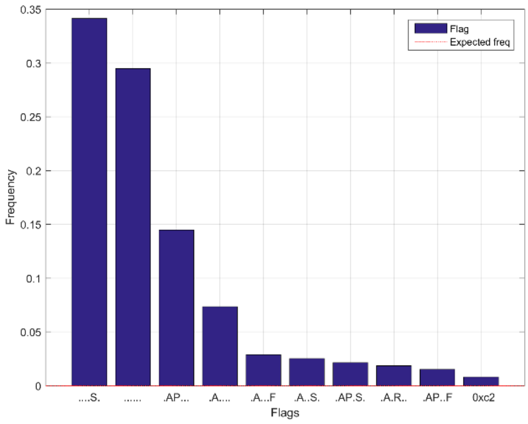


Figure 9. The statistical distribution of connection flags (TCP SYN-flood).

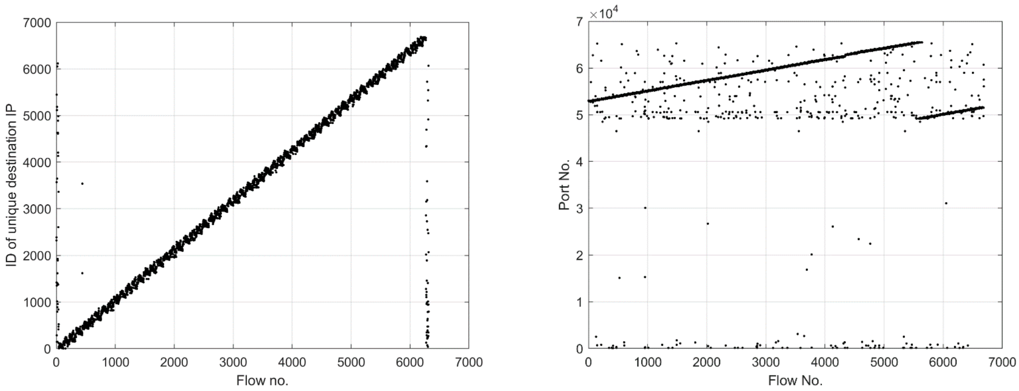


Figure 10. Unique IPs versus flow number and Ports addressed during the Scan spread attack.

The values of the time-slice-based connection features are presented in Figure 11. Note the sharp changes and peaks in the values, which may be indicative of network attacks. The statistical distribution of feature values is highly skewed and shows a considerable difference between the values. We used the violin plot, which was already used to visualize the distribution of network-traffic features before, in [58]. See the violin plot of feature-value distribution presented in Figure 12.

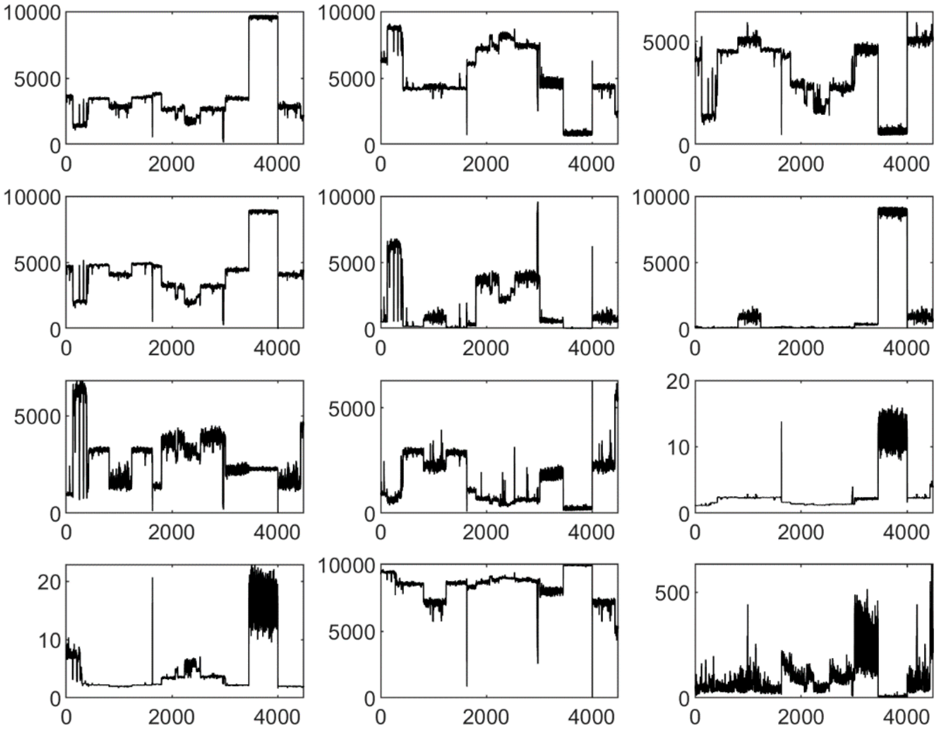


Figure 11. Values of time-slice connection features (from Table 7).

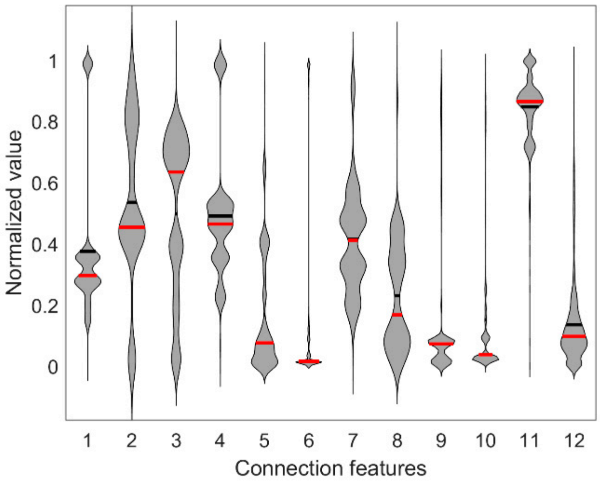


Figure 12. Statistical distribution of time-slice-based connection feature values from Table 7.

For the analysis and unsupervised clustering, we first performed the normalization of the dataset features. For the analysis of unsupervised datasets, the dimensionality reduction methods, such as Principal Component Analysis (PCA), are often used [77]. Here, we applied the t-stochastic Neighbor Embedding (t-SNE) method [78] to reduce the dimensionality to two dimensions. The resulting low-dimensional embedding has clusters assigned by the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm [79]. As a result, we obtained 12 clusters corresponding to different network-attack types, which are shown in Figure 13.

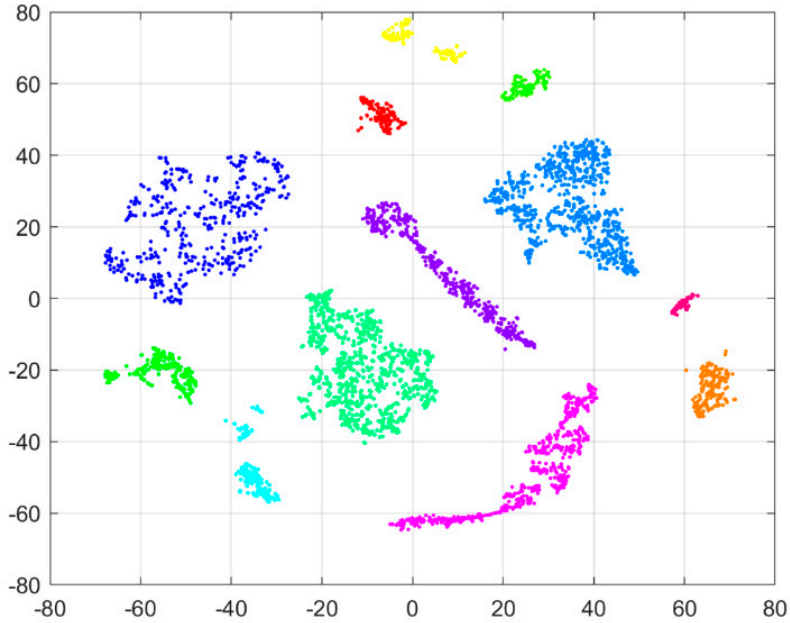


Figure 13. The results of density-based clustering of low-dimensional feature embedding obtained by using the t-stochastic Neighbor Embedding (t-SNE) method.

To analyze the differences between clusters, we adopted a pairwise two-sample multivariate Kolmogorov–Smirnov (KS) test, which is used to determine whether two sets of data arise from the same or different distributions. The null hypothesis was that the data in both pairs of compared clusters are drawn from the same continuous distribution. We used a two-dimensional version of the KS test because the low-dimensional embedding was two-dimensional. The hypothesis was rejected ($p < 0.001$) for all pairs of clusters.

The distribution of feature values according to the attack-behavior clusters in time can be seen in Figure 14. To allow for better comparison, all feature values were normalized to (0,1).

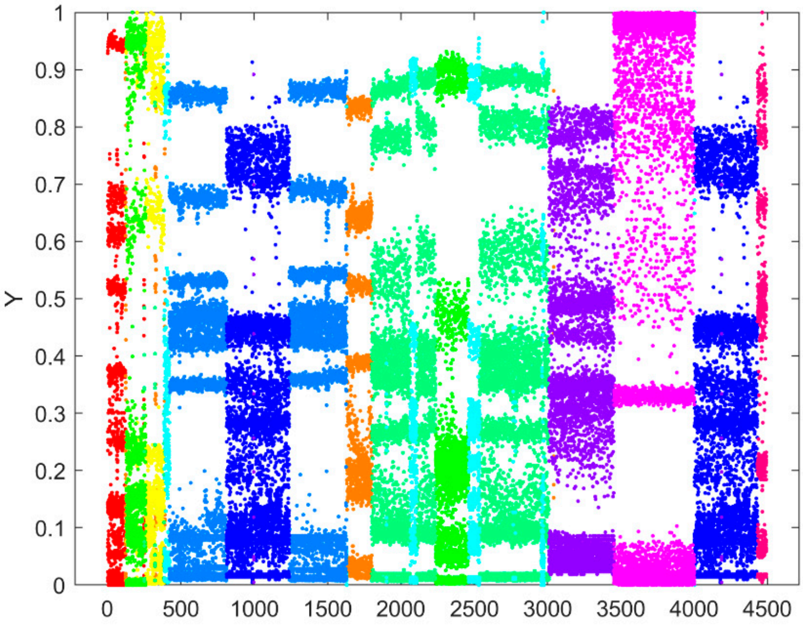
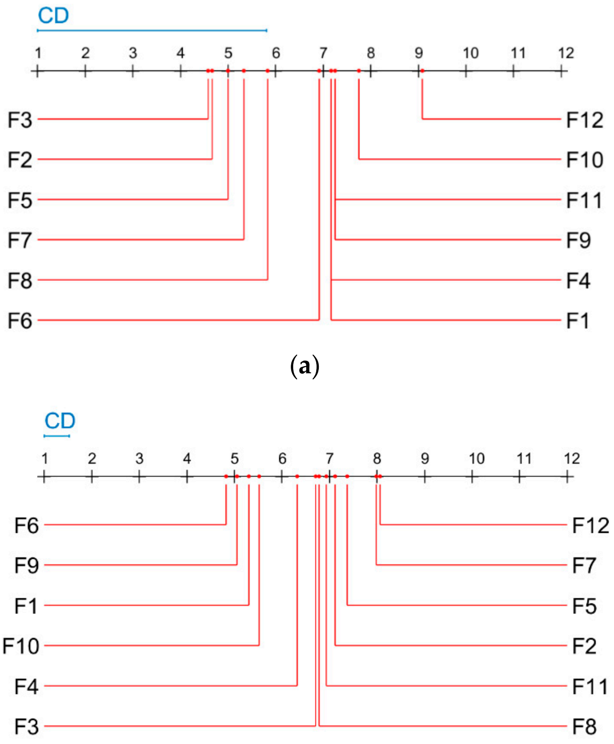


Figure 14. Distribution of time-slice feature values in time. The *y*-axis shows the normalized values of features.

To evaluate the significance of each feature, we used two tests. First, we applied the t-test-based feature, ranking using each cluster vs. all other clusters as a dependent variable. In another test, we used the split-half approach by splitting the set of clusters in half randomly and performing feature ranking, while the procedure was repeated N (N = 1000) times. The results of feature ranking were analyzed by using the non-parametric ranking-based Friedman test, and the result was statistically significant ($p < 0.05$). Finally, the post hoc Nemenyi test was applied, and its results were presented by using the significance diagram [80] (see Figure 15). Note that, according to the one-versus-all splitting, there is no statistically significant difference between the feature ranks (Figure 15a); therefore, all features are significant and contribute to the constructions of clusters. We also performed the split-half testing. The feature-ranking results (Figure 15b) show that features F6, F9, F1, and F10 have the highest rank among all features, which is statistically significant ($p < 0.001$).



(**a**)

(**b**)

Figure 15. Results of the Nemenyi test evaluating 12 time-slice-based connection features: one vs. all (**a**) and split-half (**b**).

4.4. Conclusions of Dataset Analysis

The statistical analysis of the features included in the proposed dataset shows that the features can be to for detecting various types of network attacks. In particular, the time slice connection features can be embedded to lower dimensional space, where clustering methods can be applied to map the low-dimensional representation of features to network attack classes.

5. Comparison with Other Datasets

Table 9 shows a comparative analysis with the analyzed datasets, according to the number of networks, number of unique IP address, period of the data collection, attack vectors, and the number of features for each dataset.

The proposed network dataset was collected for a longer period of time (10 months) than other analyzed datasets; it covers more network-attack classes (12) and contains more features (85). Therefore, the proposed dataset could present a valuable contribution to the research community and enrich the available set of datasets for the development and improvement of new network-attack recognition methods.

Table 9. Comparison of benchmark network datasets.

Para-meter	**DDoS 2016 [43]**	**UNSW-NB15 [59]**	**CICIDS 2017 [60]**	**UGR'16 [61]**	**NSL-KDD [38]**	**CSE-CIC-IDS2018 [62]**	**Pro-Posed Dataset**
No. of network nodes	Network simulator (NS2)	3	2	10 M different (sub)networks	n/a	5 subnets	1,395,951 IPs
Number of unique IPs	n/a	45	21	higher than 600 M	n/a	500	7,394,481
Period of data collection	n/a	15/16 h	5 days	calibration set —100 days, test set—1 month	7 weeks	17 days	10 months
Attack classes	5	9	7	7	4	7	12
No. of features	27	49	80	12	41	80	85

6. Conclusions

Known network-intrusion benchmark datasets usually do not provide the realistic case of the modern network-traffic and network-attack scenarios. In contrast, the proposed dataset contains real-world network traffic data and annotated attack examples, rather than artificially simulated attacks executed in the sandbox network environment.

To facilitate the improvement of existing network-intrusion-detection methods, and the development of new, we have suggested a new network flow dataset. The dataset has 85 features that can be used to recognize 12 different types of network attacks. We provided an analysis and comparison of the proposed dataset with two classical and four other modern datasets by key features and described its advantages and limitations. Our dataset contains real network traffic captured over 10 months. This provides an advantage over synthetically generated datasets, because an artificial synthesis of network traffic might lead to incorrect network-attack models and behaviors.

In the future, we expect that the proposed dataset can be helpful to researchers working the cybersecurity domain and can be used as a modern benchmark network-intrusion dataset.

Author Contributions: Conceptualization, R.D., A.V., and S.G.; methodology, R.D., S.G., and J.T.; software, S.G., T.A., and P.S.; validation, J.T. and N.M.; formal analysis, R.D. and J.T.; investigation, J.T. and N.M.; resources, A.V. and S.G.; data curation, S.G., T.A., and P.S.; writing—original draft preparation, R.D.; writing—review and editing, A.V., S.G., and J.T.; visualization, R.D., S.G., and J.T.; supervision, A.V.; funding acquisition, A.V. All authors have read and agreed to the published version of the manuscript.

Funding: This paper was supported, in part, by the European Union's Horizon 2020 research and innovation program, under Grant Agreement No. 830892, project "Strategic programs for advanced research and technology in Europe" (SPARTA).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Laštovička, M.; Čeleda, P. Situational Awareness: Detecting Critical Dependencies and Devices in a Network. In *Security of Networks and Services in an All-Connected World*; AIMS 2017, Zurich, Switzerland, 10–13 July 2017, Lecture Notes in Computer Science; Tuncer, D., Koch, R., Badonnel, R., Stiller, B., Eds.; Springer: Cham, Switzerland, 2017; Volume 10356, pp. 173–178.
2. Liu, B.; Bi, J.; Vasilakos, A.V. Toward Incentivizing Anti-Spoofing Deployment. *IEEE Trans. Inf. Forensics Secur.* **2014**, *9*, 436–450. [CrossRef]
3. Yao, G.; Bi, J.; Vasilakos, A.V. Passive IP Traceback: Disclosing the Locations of IP Spoofers from Path Backscatter. *IEEE Trans. Inf. Forensics Secur.* **2015**, *10*, 471–484. [CrossRef]
4. Luo, H.; Chen, Z.; Li, J.; Vasilakos, A.V. Preventing Distributed Denial-of-Service Flooding Attacks With Dynamic Path Identifiers. *IEEE Trans. Inf. Forensics Secur.* **2017**, *12*, 1801–1815. [CrossRef]

5. Jang-Jaccard, J.; Nepal, S. A survey of emerging threats in cybersecurity. *J. Comput. Syst. Sci.* **2014**, *80*, 973–993. [CrossRef]
6. Venčkauskas, A.; Morkevicius, N.; Jukavičius, V.; Damaševičius, R.; Toldinas, J.; Grigaliūnas, S. An edge-fog secure self-authenticable data transfer protocol. *Sensors* **2019**, *19*, 3612. [CrossRef] [PubMed]
7. Jing, Q.; Vasilakos, A.V.; Wan, J.; Lu, J.; Qiu, D. Security of the Internet of Things: Perspectives and challenges. *Wirel. Netw.* **2014**, *20*, 2481–2501. [CrossRef]
8. Zhou, J.; Cao, Z.; Dong, X.; Vasilakos, A.V. Security and Privacy for Cloud-Based IoT: Challenges. *IEEE Commun. Mag.* **2017**, *55*, 26–33. [CrossRef]
9. Liao, H.J.; Lin, C.H.R.; Lin, Y.C.; Tung, K.Y. Intrusion detection system: A comprehensive review. *J. Netw. Comput. Appl.* **2013**, *36*, 16–24. [CrossRef]
10. Azeez, N.A.; Ayemobola, T.J.; Misra, S.; Maskeliūnas, R.; Damaševičius, R. Network intrusion detection with a hashing based apriori algorithm using Hadoop MapReduce. *Computers* **2019**, *8*, 86. [CrossRef]
11. Nisioti, A.; Mylonas, A.; Yoo, P.D.; Katos, V. From intrusion detection to attacker attribution: A comprehensive survey of unsupervised methods. *IEEE Commun. Surv. Tutor.* **2018**, *20*, 3369–3388. [CrossRef]
12. Wazid, M.; Das, A.K.; Bhat, K.V.; Vasilakos, A.V. LAM-CIoT: Lightweight authentication mechanism in cloud-based IoT environment. *J. Netw. Comput. Appl.* **2020**, *150*, 102496. [CrossRef]
13. Yu, Y.; Xue, L.; Au, M.H.; Susilo, W.; Ni, J.; Zhang, Y.; Vasilakos, A.V.; Shen, J. Cloud data integrity checking with an identity-based auditing mechanism from RSA. *Future Gener. Comput. Syst.* **2016**, *62*, 85–91. [CrossRef]
14. Wei, W.; Woźniak, M.; Damaševičius, R.; Fan, X.; Li, Y. Algorithm Research of Known-plaintext Attack on Double Random Phase Mask Based on WSNs. *J. Internet Technol.* **2019**, *20*, 39–48. [CrossRef]
15. Challa, S.; Das, A.K.; Odelu, V.; Kumar, N.; Kumari, S.; Khan, M.K.; Vasilakos, A.V. An efficient ECC-based provably secure three-factor user authentication and key agreement protocol for wireless healthcare sensor networks. *Comput. Electr. Eng.* **2018**, *69*, 534–554. [CrossRef]
16. Khan, K.; Mehmood, A.; Khan, S.; Khan, M.A.; Iqbal, Z.; Mashwani, W.K. A survey on intrusion detection and prevention in wireless ad-hoc networks. *J. Syst. Archit.* **2020**, *105*, 101701. [CrossRef]
17. Wu, W.; Li, R.; Xie, G.; An, J.; Bai, Y.; Zhou, J.; Li, K. A survey of intrusion detection for in-vehicle networks. *IEEE Trans. Intell. Transp. Syst.* **2020**, *21*, 919–933. [CrossRef]
18. Hande, Y.; Muddana, A. A survey on intrusion detection system for software defined networks (SDN). *Int. J. Bus. Data Commun. Netw.* **2020**, *16*, 28–47. [CrossRef]
19. Shu, Z.; Wan, J.; Li, D.; Lin, J.; Vasilakos, A.V.; Imran, M. Security in Software-Defined Networking: Threats and Countermeasures. *Mob. Netw. Appl.* **2016**, *21*, 764–776. [CrossRef]
20. Li, Y.; Xu, Y.; Liu, Z.; Hou, H.; Zheng, Y.; Xin, Y.; Cui, L. Robust detection for network intrusion of industrial IoT based on multi-CNN fusion. *Meas. J. Int. Meas. Confed.* **2020**, *154*. [CrossRef]
21. Farivar, F.; Haghighi, M.S.; Jolfaei, A.; Alazab, M. Artificial Intelligence for Detection, Estimation, and Compensation of Malicious Attacks in Nonlinear Cyber-Physical Systems and Industrial IoT. *IEEE Trans. Ind. Inform.* **2020**, *16*, 2716–2725. [CrossRef]
22. Wazid, M.; Das, A.K.; Kumar, N.; Vasilakos, A.V.; Rodrigues, J.J.P.C. Design and Analysis of Secure Lightweight Remote User Authentication and Key Agreement Scheme in Internet of Drones Deployment. *IEEE Internet Things J.* **2019**, *6*, 3572–3584. [CrossRef]
23. Lin, C.; He, D.; Huang, X.; Choo, K.-K.R.; Vasilakos, A.V. BSeIn: A blockchain-based secure mutual authentication with fine-grained access control system for industry 4.0. *J. Netw. Comput. Appl.* **2018**, *116*, 42–52. [CrossRef]
24. Wazid, M.; Das, A.K.; Kumar, N.; Conti, M.; Vasilakos, A.V. A Novel Authentication and Key Agreement Scheme for Implantable Medical Devices Deployment. *IEEE J. Biomed. Health Inform.* **2018**, *22*, 1299–1309. [CrossRef] [PubMed]
25. Shalaginov, A.; Semeniuta, O.; Alazab, M. MEML: Resource-aware MQTT-based Machine Learning for Network Attacks Detection on IoT Edge Devices. In Proceedings of the 12th IEEE/ACM International Conference on Utility and Cloud Computing Companion—UCC '19 Companion, Auckland, New Zealand, 2–5 December 2019. [CrossRef]
26. Zhou, J.; Dong, X.; Cao, Z.; Vasilakos, A.V. Secure and Privacy Preserving Protocol for Cloud-Based Vehicular DTNs. *IEEE Trans. Inf. Forensics Secur.* **2015**, *10*, 1299–1314. [CrossRef]
27. Yan, Z.; Zhang, P.; Vasilakos, A.V. A security and trust framework for virtualized networks and software-defined networking. *Secur. Commun. Netw.* **2015**, *9*, 3059–3069. [CrossRef]

28. Wazid, M.; Das, A.K.; Kumar, N.; Vasilakos, A.V. Design of secure key management and user authentication scheme for fog computing services. *Future Gener. Comput. Syst.* **2019**, *91*, 475–492. [CrossRef]
29. Odusami, M.; Abayomi-Alli, O.; Misra, S.; Shobayo, O.; Damasevicius, R.; Maskeliunas, R. Android Malware Detection: A Survey. In Applied Informatics—First International Conference, ICAI 2018, Bogotá, Colombia, 1–3 November 2018. *Commun. Comput. Inf. Sci.* **2018**, *942*, 255–266. [CrossRef]
30. Rajagopal, S.; Kundapur, P.P.; Hareesha, K.S. A stacking ensemble for network intrusion detection using heterogeneous datasets. *Secur. Commun. Netw.* **2020**. [CrossRef]
31. Odusami, M.; Misra, S.; Adetiba, E.; Abayomi-Alli, O.; Damasevicius, R.; Ahuja, R. An Improved Model for Alleviating Layer Seven Distributed Denial of Service Intrusion on Webserver. *J. Phys. Conf. Ser.* **2019**, *1235*, 012020. [CrossRef]
32. Bhuyan, M.H.; Bhattacharyya, D.K.; Kalita, J.K. Network Anomaly Detection: Methods, Systems and Tools. *IEEE Commun. Surv. Tutor.* **2014**, *16*, 303–336. [CrossRef]
33. Khraisat, A.; Gondal, I.; Vamplew, P.; Kamruzzaman, J. Survey of intrusion detection systems: Techniques, datasets and challenges. *Cybersecurity* **2019**, *2*, 20. [CrossRef]
34. Alhaj, T.A.; Siraj, M.M.; Zainal, A.; Elshoush, H.T.; Elhaj, F. Feature Selection Using Information Gain for Improved Structural-Based Alert Correlation. *PLoS ONE* **2016**, *11*, e0166017. [CrossRef]
35. Ramaki, A.A.; Atani, R.E. A survey of IT early warning systems: Architectures, challenges, and solutions. *Secur. Commun. Netw.* **2016**, *9*, 4751–4776. [CrossRef]
36. Divekar, A.; Parekh, M.; Savla, V.; Mishra, R.; Shirole, M. Benchmarking datasets for anomaly-based network intrusion detection: KDD CUP 99 alternatives. In Proceedings of the 2018 IEEE 3rd International Conference on Computing, Communication and Security, ICCCS 2018, Katmandu, Nepal, 25–27 October 2018; pp. 1–8. [CrossRef]
37. Siddique, K.; Akhtar, Z.; Aslam Khan, F.; Kim, Y. KDD Cup 99 Data Sets: A Perspective on the Role of Data Sets in Network Intrusion Detection Research. *Computer* **2019**, *52*, 41–51. [CrossRef]
38. Tavallaee, M.; Bagheri, E.; Lu, W.; Ghorbani, A. A detailed analysis of the KDD CUP 99 data set. In Proceedings of the Second IEEE Symposium on Computational Intelligence for Security and Defence Applications, Ottawa, ON, Canada, 8–10 July 2009; pp. 1–6. [CrossRef]
39. Elkhadir, Z.; Mohammed, B. A cyber network attack detection based on GM median nearest neighbors LDA. *Comput. Secur.* **2019**, *86*, 63–74. [CrossRef]
40. Gao, L.; Li, Y.; Zhang, L.; Lin, F.; Ma, M. Research on detection and defense mechanisms of DoS attacks based on BP neural network and game theory. *IEEE Access* **2019**, *7*, 43018–43030. [CrossRef]
41. Yao, H.; Fu, D.; Zhang, P.; Li, M.; Liu, Y. MSML: A novel multilevel semi-supervised machine learning framework for intrusion detection system. *IEEE Internet Things J.* **2019**, *6*, 1949–1959. [CrossRef]
42. Yao, H.; Wang, Q.; Wang, L.; Zhang, P.; Li, M.; Liu, Y. An intrusion detection framework based on hybrid multi-level data mining. *Int. J. Parallel Program.* **2019**, *47*, 740–758. [CrossRef]
43. Alkasassbeh, M.; Al-Naymat, G.; Hassanat, A.; Almseidin, M. Detecting Distributed Denial of Service Attacks Using Data Mining Techniques. *Int. J. Adv. Comput. Sci. Appl. (IJACSA)* **2016**, *7*, 436–445. [CrossRef]
44. Creech, G.; Hu, J. Generation of a new IDS test dataset: Time to retire the KDD collection. In Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC), Shanghai, China, 7–10 April 2013; pp. 4487–4492. [CrossRef]
45. Koroniotis, N.; Moustafa, N.; Sitnikova, E.; Turnbull, B. Towards the development of realistic botnet dataset in the Internet of Things for network forensic analytics: Bot-IoT dataset. *Future Gener. Comput. Syst.* **2019**, *100*, 779–796. [CrossRef]
46. Bhattacharya, S.; Selvakumar, S. SSENet-2014 dataset: A dataset for detection of multiconnection attacks. In Proceedings of the 3rd International Conference on Eco-Friendly Computing and Communication Systems, ICECCS 2014, Mangalore, India, 18–21 December 2014; pp. 121–126. [CrossRef]
47. Cordero, C.G.; Vasilomanolakis, E.; Milanov, N.; Koch, C.; Hausheer, D.; Muhlhauser, M. ID2T: A DIY dataset creation toolkit for intrusion detection systems. In Proceedings of the 2015 IEEE Conference on Communications and Network Security, CNS 2015, Florence, Italy, 28–30 September 2015; pp. 739–740. [CrossRef]
48. Singh, R.; Kumar, H.; Singla, R.K. A reference dataset for network traffic activity based intrusion detection system. *Int. J. Comput. Commun. Control* **2015**, *10*, 390–402. [CrossRef]

49. Belenko, V.; Krundyshev, V.; Kalinin, M. Synthetic datasets generation for intrusion detection in VANET. In Proceedings of the 11th International Conference on Security of Information and Networks, Cardiff, UK, 10–12 September 2018; pp. 1–6. [CrossRef]
50. Vasilomanolakis, E.; Cordero, C.G.; Milanov, N.; Mühlhäuser, M. Towards the creation of synthetic, yet realistic, intrusion detection datasets. In Proceedings of the 2016 IEEE/IFIP Network Operations and Management Symposium, NOMS 2016, Istanbul, Turkey, 25–29 April 2016; pp. 1209–1214. [CrossRef]
51. Magán-Carrión, R.; Urda, D.; Díaz-Cano, I.; Dorronsoro, B. Towards a Reliable Comparison and Evaluation of Network Intrusion Detection Systems Based on Machine Learning Approaches. *Appl. Sci.* **2020**, *10*, 1775. [CrossRef]
52. Casas, P.; Mazel, J.; Owezarski, P. Unsupervised network intrusion detection systems: Detecting the unknown without knowledge. *Comput. Commun.* **2012**, *35*, 772–783. [CrossRef]
53. Kanda, Y.; Fontugne, R.; Fukuda, K.; Sugawara, T. ADMIRE: Anomaly detection method using entropy based PCA with three-step sketches. *Comput. Commun.* **2013**, *36*, 575–588. [CrossRef]
54. Meira, J.; Andrade, R.; Praça, I.; Carneiro, J.; Bolón-Canedo, V.; Alonso-Betanzos, A.; Marreiros, G. Performance evaluation of unsupervised techniques in cyber-attack anomaly detection. *J. Ambient Intell. Humaniz. Comput.* **2019**. [CrossRef]
55. Umer, M.F.; Sher, M.; Bi, Y. A two-stage flow-based intrusion detection model for next-generation networks. *PLoS ONE* **2018**, *13*. [CrossRef]
56. Fadlullah, Z.M.; Taleb, T.; Vasilakos, A.V.; Guizani, M.; Kato, N. DTRAB: Combating Against Attacks on Encrypted Protocols Through Traffic-Feature Analysis. *IEEE/ACM Trans. Netw.* **2010**, *18*, 1234–1247. [CrossRef]
57. Zhang, J.; Chen, C.; Xiang, Y.; Zhou, W.; Vasilakos, A.V. An Effective Network Traffic Classification Method with Unknown Flow Detection. *IEEE Trans. Netw. Serv. Manag.* **2013**, *10*, 133–147. [CrossRef]
58. Ring, M.; Wunderlich, S.; Scheuring, D.; Landes, D.; Hotho, A. A survey of network-based intrusion detection data sets. *Comput. Secur.* **2019**, *86*, 147–167. [CrossRef]
59. Moustafa, N.; Slay, J. UNSW-NB15: A Comprehensive Data Set for Network Intrusion Detection Systems. In *Military Communications and Information Systems Conference (MilCIS)*; IEEE: Canberra, Australia, 2015; pp. 1–6.
60. Sharafaldin, I.; Habibi Lashkari, A.; Ghorbani, A.A. A Detailed Analysis of the CICIDS2017 Data Set. In *ICISSP*; Revised Selected Papers; Springer: Cham, Switzerland, 2018; pp. 172–188.
61. Maciá-Fernández, G.; Camacho, J.; Magán-Carrión, R.; García-Teodoro, P.; Therón, R. UGR'16: A New Dataset for the Evaluation of Cyclostationarity-Based Network IDSs. *Comput. Secur.* **2018**, *73*, 411–424. [CrossRef]
62. UNB CSE-CIC-IDS2018 on AWS. Available online: https://www.unb.ca/cic/datasets/ids-2018.html (accessed on 9 May 2020).
63. Krundyshev, V.M. Preparing datasets for training in a neural network system of intrusion detection in industrial systems. *Autom. Control Comput. Sci.* **2019**, *53*, 1012–1016. [CrossRef]
64. Almomani, I.; Al-Kasasbeh, B.; AL-Akhras, M. WSN-DS: A Dataset for Intrusion Detection Systems in Wireless Sensor Networks. *J. Sens.* **2016**, *2016*, 1–16. [CrossRef]
65. Al-Hadhrami, Y.; Hussain, F.K. Real time dataset generation framework for intrusion detection systems in IoT. *Future Gener. Comput. Syst.* **2020**, *108*, 414–423. [CrossRef]
66. Zago, M.; Gil Pérez, M.; Martínez Pérez, G. UMUDGA: A dataset for profiling algorithmically generated domain names in botnet detection. *Data Brief* **2020**, *30*. [CrossRef]
67. Ahmed, M.; Naser Mahmood, A.; Hu, J. A survey of network anomaly detection techniques. *J. Netw. Comput. Appl.* **2016**, *60*, 19–31. [CrossRef]
68. Moustafa, N.; Hu, J.; Slay, J. A holistic review of network anomaly detection systems: A comprehensive survey. *J. Netw. Comput. Appl.* **2019**, *128*, 33–55. [CrossRef]
69. Salo, F.; Injadat, M.; Nassif, A.B.; Shami, A.; Essex, A. Data mining techniques in intrusion detection systems: A systematic literature review. *IEEE Access* **2018**, *6*, 56046–56058. [CrossRef]
70. DARPA Intrusion Detection Evaluation Dataset. MIT Lincoln Lab. Available online: https://www.ll.mit.edu/r-d/datasets/1998-darpa-intrusion-detection-evaluation-dataset (accessed on 9 May 2020).
71. KDD Cup 1999. Available online: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html (accessed on 9 May 2020).

72. Moustafa, N.; Slay, J. The evaluation of Network Anomaly Detection Systems: Statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set. *Inf. Secur. J. A Glob. Perspect.* **2016**, *25*, 18–31. [CrossRef]
73. Smith, R.; Japkowicz, N.; Dondo, M.; Mason, P. Using unsupervised learning for network alert correlation Advances in Artificial Intelligence. In Proceedings of the Canadian Conference on AI 2008, Windsor, Canada, 28–30 May 2008; Springer: Berlin/Heidelberg, Germany, 2008; pp. 308–319.
74. Bhuyan, M.H.; Bhattacharyya, D.K.; Kalita, J.K. Towards generating real-life datasets for network intrusion detection. *Int. J. Netw. Secur.* **2015**, *17*, 683–701.
75. Hafeez, I.; Antikainen, M.; Ding, A.Y.; Tarkoma, S. IoT-KEEPER: Detecting Malicious IoT Network Activity using Online Traffic Analysis at the Edge. *IEEE Trans. Netw. Serv. Manag.* **2020**, *17*, 45–59. [CrossRef]
76. Wahid, A.; Leckie, C.; Zhou, C. Estimating the number of hosts corresponding to an intrusion alert while preserving privacy. *J. Comput. Syst. Sci.* **2014**, *80*, 502–519. [CrossRef]
77. Eid, H.F.; Darwish, A.; Hassanien, A.E.; Abraham, A. Principle Components Analysis and Support Vector Machine based Intrusion Detection System. In Proceedings of the 10th International Conference on Intelligent Systems Design and Applications ISDA, Cairo, Egypt, 29 November–1 December 2010; pp. 363–367.
78. Van der Maaten, L.; Hinton, G. Visualizing Data using t-SNE. *J. Mach. Learn. Res.* **2008**, *9*, 2579–2605.
79. Ester, M.; Kriegel, H.-P.; Sander, J.; Xu, X. A density-based algorithm for discovering clusters in large spatial databases with noise. In Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining (KDD-96), Portland, OR, USA, 2–4 August 1996; AAAI Press: Menlo Park, CA, 1996; pp. 226–231.
80. Demšar, J. Statistical comparisons of classifiers over multiple data sets. *J. Mach. Learn. Res.* **2006**, *7*, 1–30.