sensors MDPI

Article

Utilising Flow Aggregation to Classify Benign Imitating Attacks

Hanan Hindy [1], Robert Atkinson [2], Christos Tachtatzis [2], Ethan Bayne [1], Miroslav Bures [3] and Xavier Bellekens [2,*]

[1] Division of Cybersecurity, Abertay University, Dundee DD1 1HG, UK; hananhindy@ieee.org (H.H.); e.bayne@abertay.ac.uk (E.B.)
[2] Electronic and Electrical Engineering Department, University of Strathclyde, Glasgow G1 1XQ, UK; robert.atkinson@strath.ac.uk (R.A.); christos.tachtatzis@strath.ac.uk (C.T.)
[3] Department of Computer Science, Faculty of Electrical Engineering, Czech Technical University in Prague, Karlovo Namesti 13, 121 35 Praha 2, Czech Republic; buresm3@fel.cvut.cz
* Correspondence: xavier.bellekens@strath.ac.uk

Abstract: Cyber-attacks continue to grow, both in terms of volume and sophistication. This is aided by an increase in available computational power, expanding attack surfaces, and advancements in the human understanding of how to make attacks undetectable. Unsurprisingly, machine learning is utilised to defend against these attacks. In many applications, the choice of features is more important than the choice of model. A range of studies have, with varying degrees of success, attempted to discriminate between benign traffic and well-known cyber-attacks. The features used in these studies are broadly similar and have demonstrated their effectiveness in situations where cyber-attacks do not imitate benign behaviour. To overcome this barrier, in this manuscript, we introduce new features based on a higher level of abstraction of network traffic. Specifically, we perform flow aggregation by grouping flows with similarities. This additional level of feature abstraction benefits from cumulative information, thus qualifying the models to classify cyber-attacks that mimic benign traffic. The performance of the new features is evaluated using the benchmark CICIDS2017 dataset, and the results demonstrate their validity and effectiveness. This novel proposal will improve the detection accuracy of cyber-attacks and also build towards a new direction of feature extraction for complex ones.

Keywords: NetFlow; network traffic; intrusion detection; machine learning; features; CICIDS2017; cyber-attacks

check for updates

Citation: Hindy, H.; Atkinson, R.; Tachtatzis, C.; Bayne, E.; Bures, M.; Bellekens, X. Utilising Flow Aggregation to Classify Benign Imitating Attacks. *Sensors* **2021**, *21*, 1761. https://doi.org/10.3390/s21051761

Academic Editor: Jun Zhao

Received: 4 February 2021
Accepted: 22 February 2021
Published: 4 March 2021

1. Introduction

Internet traffic pattern analysis is an established and mature discipline. A key application of this research domain is the detection of malicious network intrusions. Intrusion Detection Systems (IDS) research started in the late 1980s. Statistical models were the first to be introduced. Later, signature-based IDSs were proposed [1]. Their detection depended on known patterns (signatures) that were used to distinguish between malicious and benign traffic flows. More recently, the volume and sophistication of these attacks has increased, and that has motivated the use of machine learning (ML) techniques to counteract them [2].

In many machine learning applications, it is well known that the choice of features, i.e., the inputs fed into the model, is more important than the choice of the model [3]. Indeed, Ghaffarian and Shahriari state that features play a vital role in the development of IDS [4]. When Internet traffic is analysed on a flow-by-flow basis, the choice of features is quite self-evident: are particular flags set or reset in the internet packet headers, what is the average size of a packet in a flow, what is the standard deviation of packet sizes, what is the average time between packets in a flow, etc. In simple terms, values are extracted from packet headers, and various statistics are extracted from the packet lengths and inter-arrival times [5]. These features have proved adequate for identifying previous generations of cyber-attacks using a range of ML models; however, they have proven to be inadequate to the latest, most sophisticated attacks.

To address this problem of detecting complex cyber-attacks, specifically the benign imitating ones, we propose, in this paper, an additional level of feature abstraction, named 'Flow Aggregation'. With this approach, we look at flows at a higher level of abstraction by bundling similar flows and extracting features across them. This form of aggregation permits a deeper representation of network traffic, and increases the performance of classification models, particularly for the more sophisticated attacks that attempt to closely resemble benign network traffic and hence evade detection. The proposed features are evaluated using the benchmark CICIDS2017 [6] dataset with a particular focus on the attacks that have proven most difficult to detect using well-adopted features.

The contributions of this paper are as follows:

- The introduction of a higher level of abstraction for network traffic analysis by proposing novel features to describe bundles of flows.
- Performance improvements in binary classification of cyber-attacks when these novel features are utilised, particularly for attacks that mimic benign network traffic.
- Performance improvements in multi-class classification of cyber-attacks when these novel features are utilised.
- Performance improvements in zero-day attack detection when these novel features are utilised.

The remainder of the paper is organised as follows; Section 2 outlines the related work and the required background. The methodology is explained in Section 3. Section 4 discusses the proposed feature abstraction and the conduction of experiments alongside the results. Finally, Section 5 concludes the paper.

2. Background and Related Work

As mentioned, the detection of cyber-attacks is an established research area that has leveraged a range of technologies as it has evolved over the years [7] to cope with the exponential growth of cyber-attacks [8]. A range of ML-based models have been applied to the problem, including support vector machines, artificial neural networks, and k-means clustering [9]. Despite the discriminative power of these models, many cyber-attacks still remain undetected or have low rates of detection.

Older, more well-recognised attacks have been captured in KDD Cup'99 and the NSL-KDD datasets. These can be used to train ML-based models, and in many cases achieve good results. More contemporary cyber-attacks have been recorded in the CICIDS2017 dataset [10]. Much of the research involving this dataset has considered a subset of attacks that are particularly distinctive from benign (i.e., DDoS, Port Scan, and Web attacks), and good results have been achieved. However, some classes are either (a) left undetected, due to their similar behaviour to normal traffic, thus difficult to detect, or (b) when used in detection models, their low detection accuracy is concealed in the overall accuracy due to the class imbalance problem of this dataset [11]. In this paper, we focus on those attacks that have thus far proven elusive for researchers to identify reliably, viz. DoS Slowloris and DoS SlowHTTPTest.

Many studies use performance metrics that do not take into consideration the imbalance between relatively few examples of cyber-attacks and overwhelming examples of benign traffic. For example, an 'always zero' classifier (that always predicts benign traffic) would appear to be 99% accurate if the dataset it was tested on had 99 examples of benign traffic to each example of cyber-attack traffic. Clearly, this has the potential to provide grossly misleading results since it would never detect a single attack. To address this shortcoming, in this paper, we will fully disclose the results for precision, recall, and F1-score for each class independently.

Tables 1 and 2 provide a comprehensive list of recent studies in which the CICIDS2017 dataset has been used. The tables present the published papers, the models/techniques applied, the metrics utilised to assess performance, and the concomitant results. By observing Table 1, two findings are highlighted. Firstly, some classes (SSH, DDoS, and Port Scan, for example) have received significant attention from researchers, whilst others have been largely ignored due to their poor results and their benign-like behaviour that makes their classification difficult. Secondly, the overall accuracy is much higher than the accuracy for individual classes. For example, in [12], when the authors use 1-layer Deep Neural Network (DNN), the overall multi-class classification accuracy is 96% (Table 1) while the individual classes detection accuracies are 55.9%, 95.9%, 85.4%, and 85.2% for normal, SSH, DDoS, and port scan classes, respectively (Table 1). Similar behaviour was seen when the authors used 5-layers DNN, as demonstrated in Table 1. This indicates the misleading effect of reporting the overall accuracy when dealing with imbalanced datasets.

Table 1. CICIDS2017 recent papers performance summary (1).

Year	Ref	Approach	Covered Attacks	Accuracy	Precision	Recall	F-Score
2020	[13] +	MLP	SSH	-	82%	98%	90%
		LSTM		-	97%	98%	97%
		MLP	FTP	-	93%	77%	85%
		LSTM		-	98%	99%	99%
2019	[12] +	DNN (1 Layer)	Binary	96.3%	90.8%	97.3%	93.9%
		DNN (5 Layers)		93.1%	82.7%	97.4%	89.4%
		LR		83.9%	68.5%	85%	75.8%
		NB		31.3%	30%	97.9%	45.9%
		KNN		91.0%	78.1%	96.8%	86.5%
		SVM (RBF)		79.9%	99.2%	32.8%	49.3%
		DNN (1 Layer)	Multi-class	96%	96.9%	96%	96.2%
		DNN (5 Layers)		95.6%	96.2%	95.6%	95.7%
		LR		87%	88.9%	87%	86.8%
		NB		25%	76.7%	25%	18.8%
		KNN		90.9%	94.9%	90.9%	92.2%
		SVM (RBF)		79.9%	75.7%	79.9%	72.3%
2019	[14]	AdaBoost	DDoS	81.83%	81.83%	100%	90.01%
2018	[15]	DL	PortScan	97.80%	99%	99%	99%
		SVM		69.79%	80%	70%	65%
2018	[16]	C5.0	DDoS	85.92%	86.45%	99.70%	-
		RF		86.29%	86.80%	99.63%	-
		NB		90.06%	79.99%	86.03%	-
		SVM		92.44%	79.88%	84.36	-

+: Only snippets of the results are listed in the table. DDoS: Distributed Denial of Service; MLP: Multilayer Perceptron; DL: Deep Learning; NB: Naïve Bayes; DNN: Deep Neural Network; RBF: Radial Basis Function; FTP: File Transfer Protocol; RF: Random Forest; KNN: k-Nearest Neighbour; SSH: Secure Shell; LR: Logistic Regression; SVM: Support Vector Machine; LSTM: Long short-term memory.

Furthermore, Vinayakumar et al. [12] highlighted in their recent research that by observing the saliency map for the CICIDS2017 dataset, it is shown that "the dataset requires few more additional features to classify the connection record correctly" [12]. The authors' observations highlighted this need for the Denial of Service (DoS) class specifically. As later discussed in Section 4, this concurs with our findings regarding the attack classes that require the additional abstraction level of features to be differentiated from benign traffic and other attacks.

Table 2. CICIDS2017 recent papers performance summary (2).

Year	Ref	Approach	Accuracy			
			Normal	SSH	DDoS	PortScan
2019	[12] +	DNN (1 Layer)	55.9%	95.9%	85.4%	85.2%
		DNN (5 Layers)	56.8%	95.8%	85.5%	85.5%
		LR	88.5%	98.4%	92.2%	92.6%
		NB	32.2%	75.7%	98.5%	87.9%
		KNN	90.9%	97%	99.5%	99.6%
		SVM (RBF)	79.8%	98.8%	92.9%	99%

+: Only snippets of the results are listed in the table.

2.1. Feature Engineering

In the ML domain, features are used to represent a measurable value, a characteristic or an observed phenomenon [17]. Features are informative: they are usually represented numerically; however, some can be in categorical or string format. When categorical features are used for ML, encoding is used to transform them into a numerical (ML-friendly) format.

Obtaining features can be done by construction, extraction, or selection processes, or a combination of them [18]. Feature construction creates new features by mining existing ones by finding missing relations within features. While extraction works on raw data and/or features and apply mapping functions to extract new ones. Selection works on getting a significant subset of features. This helps reduce the feature space and reduce the computational power. Feature selection can be done through three approaches, as shown in Table 3: filter, wrapper, and embedded.

Table 3. Feature selection approaches.

Approach	Description	Advantages	Disadvantages
Filter [19]	Selects the most meaningful features regardless of the model	Low Execution Time and over-fitting	May choose redundant variables
Wrapper [20]	Combine related variables to have subsets	Consider interactions	Over-fitting risk and High execution time
Embedded [21]	Investigate interaction more thoroughly than Wrapper	Result in an optimal subset of variables	–

Law et al. [22] highlight the importance of feature selection for ML models. The authors discuss its effect on boosting performance and reducing the effect of noisy features, specifically when training using small datasets. Furthermore, non-uniform class distributions should be considered to avoid misleading results when applying a supervised feature selection [23]. Alternatively, when the dataset labels are not available, an unsupervised feature selection is used. Mitra et al. [24] categorise unsupervised feature selection techniques into (a) clustering performance maximisation and (b) feature dependencies and relevance.

2.2. *Artificial Neural Network*

ANNs are inspired by the human biological brain. McCulloch and Pitts [25] proposed the first ANN in 1943. Later in 1986, Rumelhart and McClelland [26] introduced the back propagation concept. ANNs are used to estimate a complex function by learning to generalise using the given input values and the corresponding output values.

An ANN is generally composed of an input layer, zero or more hidden layers, and an output layer. Each layer is composed of one or more neurons. Neurons in layer i are connected to the ones in layer j, $j = i + 1$. This connection is called weight and is represented as w_{ij}. During the training process, the input values are propagated forward, the error is calculated (based on the expected output), then the error is propagated, and the weights are adjusted accordingly. The weight of a connection implies the significance of the input.

Formally, the output of a single neuron is calculated as shown in Equation (1).

$$O = f((\sum_{i=0}^{n} x_i.w_i) + b) \tag{1}$$

where n represents the number of inputs to this node, x_i is the i^{th} input value, w_i is the weight value, b is a bias value. Finally, f is the activation function, which squashes the output. Activation function can be, but not limited to, Tanh, Sigmoid, and Rectified Linear Unit (RELU).

The error E is calculated at the final layer using the difference between the expected output and the predicted output (which is, as aforementioned, a result of propagating the input signal). Finally, the weights are updated based on Equation (2)

$$w_{t+1} = w_t - \eta \frac{dE}{dw_t} \tag{2}$$

where w_t is the old weight and w_{t+1} is the new weight. η is the learning rate to control the gradient decent steps.

The weight of a neuron is directly proportional to the significance of the node's input. This is because the output of any neuron is calculated by multiplying the weights by the input values.

The next section will discuss the proposed features, then Section 4 outlines the experiments where ANNs are used for the classification purpose.

3. Methodology

Given a raw capture of internet traffic in the form of a raw "*pcap*" file, two levels of features are traditionally extracted. As illustrated in Figure 1, at the lower level, individual packets are inspected and packet-based features are extracted. These features include flags, packet size, payload data, source and destination address, protocol, etc. At the higher level, flow features (unidirectional and bidirectional) are extracted that consider all the individual packets in a particular communication.

In this paper, a novel additional (third) level of abstraction is proposed where bidirectional flows are grouped into bundles and flow aggregation features are derived. Flow aggregation features aim at representing information about the whole communication between network hosts. These features provide additional traffic characteristics by grouping individual flows. In this setup, it is assumed that legitimate hosts establish secure communication using any of the well-known authentication mechanisms [27,28]. After these aggregated features are calculated, they are propagated back to each bidirectional flow in the bundle/group. This is represented with the superscript + sign in Figure 1. The two proposed flow aggregation features in this manuscript are (a) the number of flows and (b) the source ports delta.

The first feature, 'Number of flows', represents the number of siblings in a flow bundle. Given the communication between a host, A, and one or more hosts, all flows initiated by A are counted. This feature represents the communication flow. The advantage of this feature is that it is significant for attacks that intentionally spread their associated requests over time when targeting a single host. However, when grouped, the bundled flow will have additional information about how many flows are in the same group that can resemble the communication pattern. Moreover, it can represent patterns when an attacker targets many hosts, but each with a few communications, when grouped, a pattern can be identified.

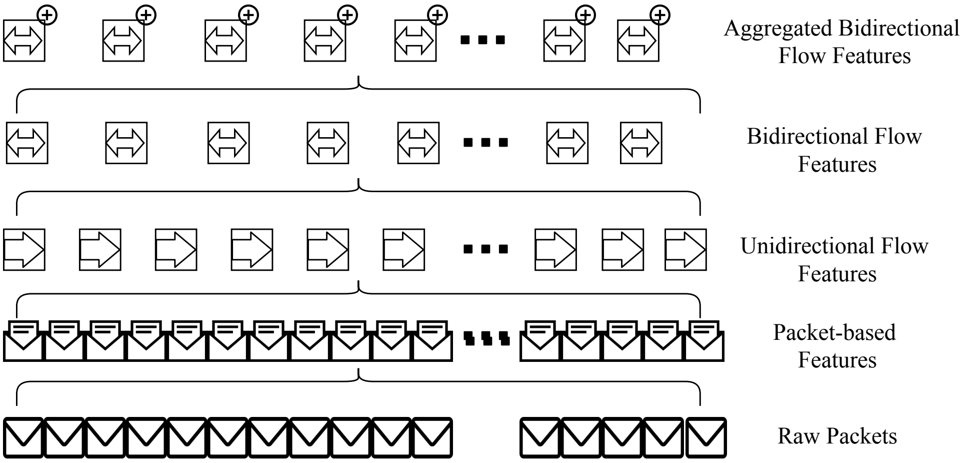

Aggregated Bidirectional Flow Features
Bidirectional Flow Features
Unidirectional Flow Features
Packet-based Features
Raw Packets

Figure 1. Networking features abstraction levels.

Figure 2 visualises the bundling process of flows. Each letter (Figure 2 top) represents a node/host in the network. Each double arrow represents a bidirectional flow with the notation XY_i, such that X is the source node, Y is the destination node, and i is the communication counter. Finally, the colours in Figure 2 represent the grouping of flows into bundles.

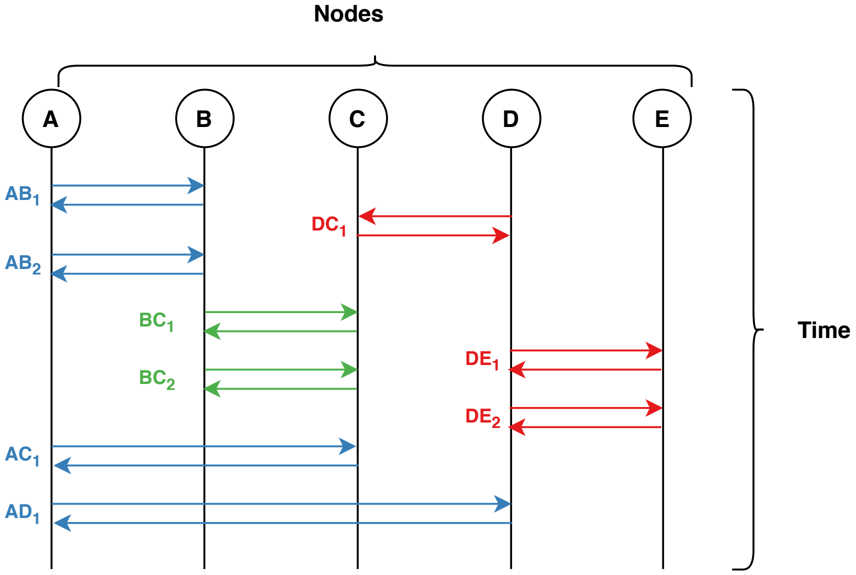


Figure 2. Flow aggregation of network traffic flows. Each colour represents an aggregated flow.

With reference to Figure 2, the first bundle (blue) has 4 flows, thus AB_1, AB_2, AC_1, and AD_1 will have the 'number of flows' feature set to 4. Similarly, the second bundle (green), BC_1 and BC_2 will have the value 2 and so on.

The second feature 'source ports delta' demonstrates the ports delta. This feature is calculated using all the port numbers used in a bundle communication flow. Algorithm 1 illustrates the feature calculation. The advantage of this feature is to capture the level and variation pattern of used ports in legitimate traffic. This feature adds this piece of information to each flow, which then enhances the learning and classification as further discussed in Section 4.

Algorithm 1 Calculate Ports Delta Feature.

```
**Input:** List of bundle flow ports
**Output:** Ports Delta Feature
ports.sort()
for $i \in length(ports) - 1$ do
    diff[i] ← abs(ports[i+1] - ports[i])
end for
avg_diff ← diff.mean()
return avg_diff
```

To validate the significance of the newly added features proposed in this manuscript, a feature selection algorithm is used. Recursive Feature Elimination (RFE) [29] is used to select the best k features to use for classification. Over the various experiments discussed in Section 4, RFE demonstrates that the two features are important for identifying classes that mimic benign behaviour. This emphasises the case in which the additional level of feature abstraction is needed. When attackers attempt to mimic benign behaviour, the attacks are in-distinctive using flow features solely. Therefore, new features are needed.

4. Experiments and Results

In this section, the conducted experiments are explained and the results are discussed. For a dataset to be appropriate to evaluate the proposed features, it has to (a) include both benign and cyber-attack traffic, (b) cover benign mimicking cyber-attacks, and (c) involve multiple attackers to guarantee the aggregation logic. Based on these criteria, the CICIDS2017 dataset [6] is selected. The dataset contains a wide range of insider and outsider attacks alongside benign activity. Sarker et al. [8] highlight in their survey that the CICIDS2017 dataset is suitable for evaluating ML-based IDS including zero-day attacks [8]. As aforementioned, the focus of this manuscript is on attacks that mimic benign behaviour.

Therefore, the attacks of interest from the CICIDS2017 dataset are (1) DoS Slowloris and (2) DoS SlowHTTPTest. These two attacks implement low-bandwidth DoS attacks in the application layer. This is done by draining concurrent connections pool [30]. Since these two attacks are performed slowly, they are typically hard to detect. Alongside the DoS SlowHTTPTest and Slowloris, two other attacks are of interest for comparative purposes: (3) PortScan and (4) DoS Hulk. These two attacks resemble the case where attacks are easy to discriminate from benign traffic.

First, each of these four attacks and benign pcap files is processed. The output of this process is bidirectional flow features and aggregation features. Second, RFE is used to select the best $k = 5$ features. Third, the selected features are used as input to an ANN that acts as the classifier. The ANN architecture is composed of 5 input neurons, 1 hidden layer with 3 neurons, and an output layer. It is important to mention that since the focus is on the evaluation of the additional level of features and not the classifier complexity, the number of chosen features is small, and a straightforward ANN is used.

Three experiments are performed. The first experiment is a binary classification of each of the attacks of interest (Section 4.2). The second experiment is a 3-class classification (Section 4.3). This experiment evaluates the classification of benign, a benign-mimicking attack, and a distinctive attack (i.e., do not mimic benign behaviour). Finally, the third experiment is a 5-class classification including all classes of interest (Section 4.4). Each experiment is performed twice, (a) with the bidirectional features only and (b) with the bidirectional features and the aggregation features. The RFE is performed independently in each experiment. It is important to highlight that the features selected by RFE confirm the importance of the proposed flow aggregation ones as follows. The top two RFE features for the 5-class classification are 'Number of Flows' and 'Source ports delta'. For the three class classification of the Slowloris, the top two RFE features are 'Source ports delta' and

'Number of Flows'. Similarly, the 'Number of Flows' feature is chosen by RFE for the 3-class classification and the 'Source ports delta' for the binary classification. This confirms the significance of the new features prior to analysing the classification results when using the additional features. For the purpose of evaluation comparison, the RFE features are selected without the flow aggregation ones in consideration.

4.1. Evaluation Metrics

In this section, the used evaluation metrics are discussed. The evaluation is performed in a 10-fold cross-validation manner. Recall, precision, and F1-score are reported for each experiment. Their formulas are shown in Equations (3)–(5), respectively. True Positive (TP) represents attack instances correctly classified, False Negative (FN) represents attack instances misclassified, and False Positive (FP) represents benign instances misclassified.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{3}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP + FP}} \tag{4}$$

$$\text{F1} = \frac{\text{2TP}}{\text{2TP + FP + FN}} \tag{5}$$

4.2. Binary Classification Results

The first experiment is a binary classifier. Each of the attacks of interest is classified against benign behaviour. Tables 4 and 5 show the precision, recall, and F1-score for DoS Slowloris and DoS SlowHTTPTest, respectively. Moreover, the table lists the five features picked by RFE. By observing the recall of the attack class with the flow aggregation features included, it can be seen that the recall rose from 83.69% to 91.31% for the Slowloris attack class and from 65.94% to 70.03% for the SlowHTTPTest attack class. Unlike benign mimicking attacks, aggregation features did not provide benefit when classifying attacks that do not mimic benign traffic such as PortScan and DoS Hulk, as shown in Table 6 and Table 7. Precision and recall were high (99%) for both of these attacks without utilising the aggregation flow features. This is coherent with the discussion in Section 2.

Finally, Figure 3 additionally visualises the effect flow aggregation features have on the recall of attack classes. It is observed that the two attack classes that mimic benign behaviour (DoS, SlowHTTPTest and Slowloris) had an increase in recall. However, for the other classes (PortScan and DoS Hulk) the aggregation did not impact the classification performance. This is due to the strength of bidirectional flow features to discriminate classes.

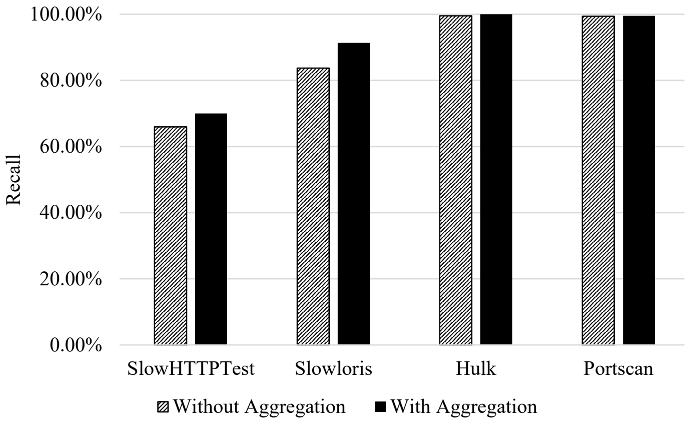


Figure 3. Binary classification | Impact of aggregation on attack class recall (benign vs. attack classification).

Table 4. Benign vs. Slowloris classification (5-fold cross validation).

	Without Aggregation			**With Aggregation**		
RFE Selected Features	- Fwd Min Inter-arrival Time - Bwd Min Inter-arrival Time - Bwd mean time between the first packet and each successive packet - Fwd mean time between the first packet and each successive packet - Fwd STD Inter-arrival Time			- Without Aggregation Features + **- Number of Flows** + **- Source Ports Delta**		
	Precision	**Recall**	**F1**	**Precision**	**Recall**	**F1**
Benign	99.04% ± 0.08%	99.86% ± 0.13%	99.45% ± 0.05%	99.49% ± 0.08%	99.99% ± 0.01%	99.74% ± 0.04%
Slowloris	97.35% ± 2.35%	83.69% ± 1.42%	89.97% ± 0.81%	99.73% ± 0.26%	91.31% ± 1.35%	95.33% ± 0.76%

Table 5. Benign vs. SlowHTTPTest classification (5-fold cross validation).

	Without Aggregation			**With Aggregation**		
RFE Selected Features	- Fwd mean time between the first packet and each successive packet - Bwd mean time between the first packet and each successive packet - Fwd Min Inter-arrival Time - Bwd Min Inter-arrival Time - Fwd Max Inter-arrival Time			- Without Aggregation Features + **- Number of Flows** + **- Source Ports Delta**		
	Precision	**Recall**	**F1**	**Precision**	**Recall**	**F1**
Benign	98.49% ± 0.04%	99.94% ± 0.02%	99.21% ± 0.03%	98.68% ± 0.40%	99.87% ± 0.14%	99.27% ± 0.17%
SlowHTTP Test	98.13% ± 0.56%	65.94% ± 0.98%	78.87% ± 0.82%	96.24% ± 3.68%	70.03% ± 9.27%	80.63% ± 5.21%

Table 6. Benign vs. DoS Hulk classification (5-fold cross validation).

	Without Aggregation			**With Aggregation**		
RFE Selected Features	- Bwd Min Packet Length - Fwd Num Reset Flags - Bwd Num Push Flags - Bwd Num Reset Flags - Fwd Max Inter-arrival Time			- Without Aggregation Features + **- Number of Flows** + **- Source Ports Delta**		
	Precision	**Recall**	**F1**	**Precision**	**Recall**	**F1**
Benign	99.83% ± 0.04%	99.99% ± 0.01%	99.91% ± 0.02%	100.00% ± 0.00%	100.00% ± 0.00%	100.00% ± 0.00%
Hulk	99.98% ± 0.03%	99.51% ± 0.10%	99.74% ± 0.06%	99.99% ± 0.02%	99.99% ± 0.02%	99.99% ± 0.02%

Table 7. Benign vs. PortScan classification (5-fold cross validation).

	Without Aggregation			**With Aggregation**		
RFE Selected Features	- Fwd STD Packet Length - Bwd Min Packet Length - Fwd Max Packet Length - Fwd Mean Packet Length - Fwd Number of Push Flags			- Without Aggregation Features + **- Number of Flows** + **- Source Ports Delta**		
	Precision	**Recall**	**F1**	**Precision**	**Recall**	**F1**
Benign	99.40% ± 0.03%	99.36% ± 0.93%	99.38% ± 0.45%	99.51% ± 0.03%	100.00% ± 0.00%	99.75% ± 0.01%
Portscan	99.36% ± 0.92%	99.39% ± 0.04%	99.37% ± 0.45%	100.00% ± 0.00%	99.50% ± 0.03%	99.75% ± 0.01%

4.3. Three-Classes Classification Results

In the second experiment, a three-class classification is performed. Benign and PortScan classes (which act as the discriminative class) are used with each of the other attack classes. Similarly, experimental results demonstrate the high recall of both benign and PortScan with and without the use of flow aggregation features and the rise in the other attack recall when flow aggregation features are used.

By observing Table 8, the recall of DoS Slowloris class increased from 78.25% to 99.09% when flow aggregation was used. Similarly, the DoS SlowHTTPTest rose from 0% to 58.97% in Table 9. Finally, DoS Hulk showed similar behaviour to the binary classification as shown in Table 10.

Figure 4 visualises the effect flow aggregation features have on the recall of attack classes in a three-class classification problem. The recall for DoS SlowHTTPTest without using the flow aggregation was 0%.

Table 8. Benign vs. PortScan vs. Slowloris classification (5-fold cross validation).

	Without Aggregation			**With Aggregation**		
RFE Selected Features	- Fwd STD Packet Length - Bwd Min Packet Length - Bwd mean time between the first packet and each successive packet - Fwd mean time between the first packet and each successive packet - Fwd Max Packet Length			- Without Aggregation Features + **- Number of Flows** + **- Source Ports Delta**		
	Precision	**Recall**	**F1**	**Precision**	**Recall**	**F1**
Benign	98.31% ± 0.16%	97.69% ± 1.02%	98.00% ± 0.49%	99.46% ± 0.04%	99.99% ± 0.01%	99.73% ± 0.02%
Portscan	97.85% ± 0.99%	99.60% ± 0.12%	98.71% ± 0.45%	100.00% ± 0.01%	99.50% ± 0.03%	99.74% ± 0.01%
Slowloris	96.95% ± 1.62%	78.25% ± 1.68%	86.59% ± 1.45%	99.75% ± 0.15%	99.09% ± 0.44%	99.42% ± 0.21%

Table 9. Benign vs. PortScan vs. SlowHTTPTest classification (5-fold cross validation).

	Without Aggregation			**With Aggregation**		
RFE Selected Features	- Fwd Mean Packet Length - Fwd STD Packet Length - Fwd Max Packet Length - Bwd mean time between the first packet and each successive packet - Fwd mean time between the first packet and each successive packet			- Without Aggregation Features + **- Number of Flows** + **- Source Ports Delta**		
	Precision	**Recall**	**F1**	**Precision**	**Recall**	**F1**
Benign	95.10% ± 0.04%	96.40% ± 0.12%	95.75% ± 0.06%	97.67% ± 1.25%	99.99% ± 0.01%	98.81% ± 0.64%
Portscan	96.45% ± 0.11%	99.52% ± 0.05%	97.96% ± 0.06%	99.99% ± 0.02%	99.42% ± 0.15%	99.70% ± 0.08%
SlowHTTP Test	0.00% ± 0.00%	0.00% ± 0.00%	0.00% ± 0.00%	79.62% ± 39.81%	58.97% ± 29.88%	67.67% ± 34.00%

Table 10. Benign vs. PortScan vs. DoS Hulk classification (5-fold cross validation).

	Without Aggregation			With Aggregation		
RFE Selected Features	- Fwd Mean Packet Length - Fwd Max Packet Length - Fwd Number of RST Flags - Fwd Number of Push Flags - Bwd Number of RST Flags			- Without Aggregation Features + **- Number of Flows** + **- Source Ports Delta**		
	Precision	**Recall**	**F1**	**Precision**	**Recall**	**F1**
Benign	98.10% ± 0.05%	99.30% ± 0.95%	98.69% ± 0.49%	99.57% ± 0.25%	99.94% ± 0.04%	99.75% ± 0.12%
Portscan	99.10% ± 0.94%	99.39% ± 0.04%	99.24% ± 0.48%	99.95% ± 0.03%	99.73% ± 0.24%	99.84% ± 0.11%
Hulk	99.94% ± 0.03%	98.56% ± 0.06%	99.25% ± 0.03%	99.98% ± 0.03%	99.50% ± 0.06%	99.74% ± 0.03%

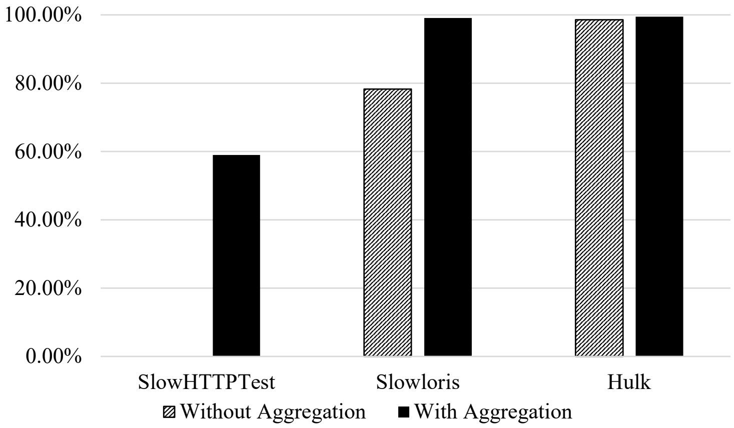


Figure 4. Multi-class classification | Impact of aggregation on the second attack class recall (benign vs. PortScan vs. attack classification).

4.4. Five-Classes Classification Results

The third experiment combines all the classes of interest in a five-class classification problem. Similarly, the experiment is performed twice—with and without the use of flow aggregation features. Table 11 summarises the performance per class for this experiment. A few observations are as follows; (a) the recall of DoS Slowloris rose from 1.40% to 67.81%. (b) The recall of DoS SlowHTTPTest rose from 0% to 4.64% only. This is not because the new features were not significant, but because the model classified DoS SlowHTTPTest as DoS Slowloris. In this case, flow aggregation features serve to discriminate benign-mimicking attacks from benign traffic but not to discriminate them from each other. Without the aggregation features, 82.84% of DoS SlowHTTPTest attack instances were classified as benign; however, this dropped to 58% when the flow aggregation features were used.

To overcome this low recall, extra features were chosen. Five features were added and the hidden layer neurons were 8. The results of this classification experiment are summarised in Table 12. The rise in the recall for the attack classes with and without flow aggregation features was as follows; from 33.94% to 80.39% and 21.45% to 64.91% for DoS Slowloris and DoS SlowHTTPTest, respectively.

This behaviour is recognised in Figure 5. It is important to mention that there was a rise in the recall of all classes. This rise was more significant for the attack classes that mimic benign behaviour than others.

Table 11. Five classes classification (5-fold cross validation).

	Without Aggregation			**With Aggregation**		
RFE Selected Features	- Fwd Mean Packet Length - Bwd Mean Inter-arrival time - Fwd mean time between the first packet and each successive packet - bwd mean time between the first packet and each successive packet - Fwd Max packet length			- Without Aggregation Features + **- Number of Flows** + **- Source Ports Delta**		
	Precision	**Recall**	**F1**	**Precision**	**Recall**	**F1**
Benign	90.97% ± 2.99%	96.77% ± 0.80%	93.74% ± 1.33%	95.11% ± 2.11%	97.11% ± 3.30%	96.05% ± 1.84%
Portscan	97.12% ± 0.80%	98.90% ± 1.05%	98.00% ± 0.38%	99.92% ± 0.13%	99.41% ± 0.15%	99.67% ± 0.08%
Slowloris	18.89% ± 37.78%	1.40% ± 2.80%	2.61% ± 5.22%	66.88% ± 8.94%	67.81% ± 31.08%	63.14% ± 25.95%
SlowHTTP Test	0.00% ± 0.00%	0.00% ± 0.00%	0.00% ± 0.00%	34.12% ± 42.81%	4.64% ± 6.51%	8.10% ± 11.21%
Hulk	93.75% ± 6.47%	98.61% ± 0.73%	95.98% ± 3.14%	93.00% ± 8.58%	99.34% ± 0.15%	95.85% ± 4.87%

Table 12. Five classes classification (5-fold cross validation).

	Without Aggregation			**With Aggregation**		
RFE Selected Features	Five RFE features + - Fwd Max Inter-arrival time - Fwd STD Inter-arrival time - Fwd Number of Reset Flags - Fwd Number of Bytes - Bwd Max Inter-arrival time			- Without Aggregation Features + **- Number of Flows** + **- Source Ports Delta**		
	Precision	**Recall**	**F1**	**Precision**	**Recall**	**F1**
Benign	92.37% ± 3.56%	96.34% ± 0.11%	94.28% ± 1.83%	97.35% ± 0.53%	99.90% ± 0.13%	98.61% ± 0.31%
Portscan	96.48% ± 0.07%	99.74% ± 0.03%	98.08% ± 0.04%	99.84% ± 0.18%	99.59% ± 0.20%	99.71% ± 0.10%
Slowloris	38.91% ± 47.65%	33.94% ± 41.60%	36.25% ± 44.41%	93.52% ± 5.64%	80.39% ± 2.66%	86.44% ± 3.94%
SlowHTTP Test	37.52% ± 45.98%	21.45% ± 26.27%	27.29% ± 33.44%	96.80% ± 2.72%	64.91% ± 16.75%	76.61% ± 12.25%
Hulk	99.92% ± 0.14%	98.63% ± 0.67%	99.27% ± 0.29%	99.88% ± 0.14%	99.69% ± 0.21%	99.78% ± 0.15%

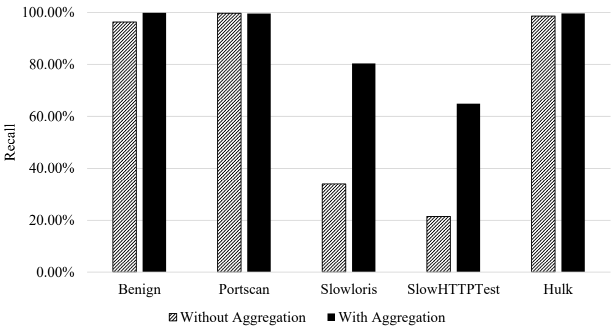


Figure 5. Multi-class Classification | Impact of Aggregation on the Classes Recall.

4.5. Zero-Day Attack Detection Revaluation

The authors' previous work in [31] proposed an autoencoder model to detect zero-day attacks. The model relies on the encoding–decoding capabilities of autoencoders to flag unknown (zero-day) attacks. The model performance was evaluated using all the CICIDS2017 dataset attack classes using three threshold values (0.15, 0.1, and 0.05). The published results demonstrate the ability of the autoencoder to effectively detect zero-day attacks; however, the attacks that mimic benign behaviour experienced very low detection rates. In this section, the published model [31] was re-evaluated using the proposed higher level of feature abstraction. The aim is to assess the impact of the new features on zero-day attack detection, specifically benign mimicking ones, whose detection rate is low.

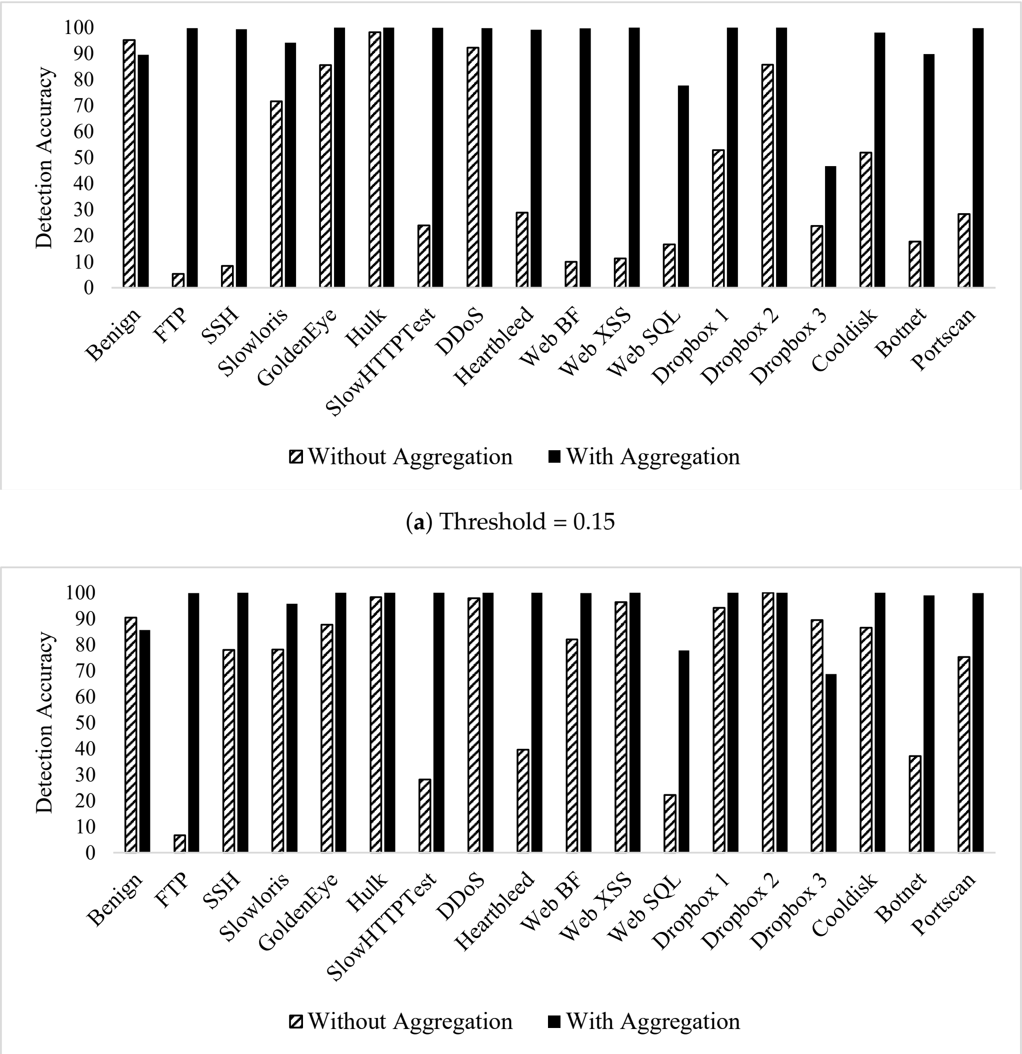
Table 13 lists the zero-day detection accuracies when flow aggregation features are used alongside the bidirectional flow ones, which were used solely in the previously published work. The results show a high detection rate of all attacks, including the attacks that were detected with low accuracy without the flow aggregation features in [31].

Table 13. CICIDS2017 autoencoder zero-day detection results using flow aggregation features.

Class	**Accuracy**		
Threshold	**0.15**	**0.1**	**0.05**
Benign (Validation)	89.5%	85.62%	67.59%
FTP Brute-force	99.81%	99.92%	100%
SSH Brute-force	99.37%	100%	100%
DoS (Slowloris)	94.12%	95.77%	100%
DoS (GoldenEye)	100%	100%	100%
DoS (Hulk)	100%	100%	100%
DoS (SlowHTTPTest)	99.91%	100%	100%
DDoS	99.79%	100%	100%
Heartbleed	99.13%	100%	100%
Web BF	99.7%	99.94%	100%
Web XSS	100%	100%	100%
Web SQL	77.78%	77.78%	100%
Infiltration—Dropbox 1	100%	100%	100%
Infiltration—Dropbox 2	100%	100%	100%
Infiltration—Dropbox 3	46.76%	68.68%	99.82%
Infiltration—Cooldisk	98.08%	100%	100%
Botnet	89.83%	98.98%	100%
Portscan	99.81%	99.85%	100%

To visualise the impact of flow aggregation features on zero-day attack detection, Figure 6 shows the effectiveness of flow aggregation features by contrasting the results that are discussed in this section versus the ones in [31]. It is observed that all attacks experienced a rise in detection accuracy when the flow aggregation features were used.

In summary, flow aggregation features prove their effectiveness in providing a deeper representation of a network traffic, thus improving classification performance. This is reflected in the evaluation of both zero-day attacks detection and multi-class attack classifier. The proposed approach has the following three limitations. (a) The unavailability of traffic flow data affects the flow aggregation features computation. (b) More features can be derived from the aggregated flows that can further improve the classification accuracy. Lastly, (c) the proposed features are evaluated using the CICIDS2017, real-time evaluation will provide additional insights.



(**a**) Threshold = 0.15

(**b**) Threshold = 0.1

Figure 6. CICIDS2017 autoencoder zero-day detection comparison using flow aggregation.

5. Conclusions

Traditional traffic features have proven powerful when combined with sufficient training examples to train ML-based classifiers, and the trained models are capable of classifying some cyber-attacks. However, some cyber-attacks are left undetected. To resolve this limitation, there are two alternatives: (a) gather huge amounts of data to be able to build even more complex models, which is difficult and impractical in some cases, and (b) represent the data using more powerful features. In this paper we appoint the second approach.

This paper presents an additional abstraction level of network flow features. The objective is to improve the cyber-attack classification performance for attacks that mimic benign behaviour. Cyber-attacks are becoming more complex, and attackers utilise the available knowledge to tailor attacks that can bypass detection tools by acting like benign traffic.

The idea is to aggregate bidirectional flows to bundles and compute bundle-specific features. Once the features are computed, the values are populated back to the bidirectional flows. The advantage of these additional features is that the bidirectional flows have some additional knowledge/information about their sibling flows.

The proposal is evaluated using the CICIDS2017 dataset. A group of attacks are used to assess the significance of the new features as well as the performance gain. Four cyber-attack classes are used beside benign class: DoS Slowloris, DoS SlowHTTPTest, DoS Hulk, and PortScan. ANN is used as the classifier. Three experiments are conducted: binary, three-class, and five-class classification. The experiments confirm the need for this additional level of features. The results further demonstrate the significance of the added features for classes that are hard to discriminate from benign, such as DoS Slowloris and DoS SlowHTTPTest. The recall of the cyber-attack classes experiences a high rise when the additional features are used. For example, it is observed that the recall of the DoS Slowloris class rises from 83.97% with the bi-directional features to 91.31 in binary classification, from 78.28% to 99.09% for 3-class classification and from 33.94% to 80.39% for 5-class classification.

Furthermore, the additional features prove significant when reassessing the authors' previously published work on detecting zero-day attacks. Benign-mimicking attacks suffered low detection accuracy; however, with the use of flow aggregation features, zero-day attack detection experience a high rise in accuracy.

Future work involves evaluating the significance of the flow aggregation against other operations alongside extracting other high-level features based on needs.

Author Contributions: Conceptualisation, H.H., R.A. and X.B.; Formal analysis, H.H., R.A. and C.T.; Investigation, H.H. and R.A.; Methodology, H.H., R.A., C.T., M.B. and X.B.; Project administration, X.B.; Software, H.H.; Supervision, E.B. and X.B.; Validation, R.A., C.T., E.B., M.B. and X.B.; Writing—original draft, H.H.; Writing—review & editing, R.A., C.T., E.B., M.B. and X.B. All authors have read and agreed to the published version of the manuscript.

Funding: The research is supported by the European Union Horizon 2020 Programme under Grant Agreement No. 833673. The content reflects the authors' view only and the Agency is not responsible for any use that may be made of the information within the paper.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The 'CICIDS2017' dataset supporting the conclusions of this article is available in the Canadian Institute for Cybersecurity (CIC) repository, http://www.unb.ca/cic/datasets/ids-2017.html (accessed on: 7 October 2019). The code will be available through a GitHub repository.

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

Abbreviations

The following abbreviations are used in this manuscript:

DDoS	Distributed Denial of Service
DoS	Denial of Service
DNN	Deep Neural Network
FN	False Negative
FP	False Positive
IDS	Intrusion Detection System
RELU	Rectified Linear Unit
RFE	Recursive Feature Elimination
ML	Machine Learning
TP	True Positive

References

1. Khraisat, A.; Gondal, I.; Vamplew, P.; Kamruzzaman, J. Survey of Intrusion Detection Systems: Techniques, Datasets and Challenges. *Cybersecurity* **2019**, *2*, 20. [CrossRef]
2. Hindy, H.; Tachtatzis, C.; Atkinson, R.; Brosset, D.; Bures, M.; Andonovic, I.; Michie, C.; Bellekens, X. Leveraging Siamese Networks for One-Shot Intrusion Detection Model. *arXiv* **2020**, arXiv:2006.15343.
3. Ravi, K.V.; Kumari, V.V.; Kumar, S.S. A Survey of Feature Selection Techniques in Intrusion Detection System: A Soft Computing Perspective. In *Progress in Computing*; Pattnaik, P.K., Rautaray, S.S., Das, H., Nayak, J., Eds.; Springer: Singapore, 2018; pp. 785–793. [CrossRef]
4. Ghaffarian, S.M.; Shahriari, H.R. Software Vulnerability Analysis and Discovery Using Machine-Learning and Data-Mining Techniques: A Survey. *ACM Comput. Surv. CSUR* **2017**, *50*, 56. [CrossRef]
5. Alaidaros, H.; Mahmuddin, M.; Al Mazari, A. An overview of flow-based and packet-based intrusion detection performance in high speed networks. In Proceedings of the International Arab Conference on Information Technology, Zarqa, Jordan, 28–30 November 2011; pp. 1–9.
6. Sharafaldin, I.; Lashkari, A.H.; Ghorbani, A.A. Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization. In Proceedings of the 4th International Conference on Information Systems Security and Privacy (ICISSP 2018), Madeira, Portugal, 22–24 January 2018; pp. 108–116.
7. Liu, H.; Lang, B. Machine Learning and Deep Learning Methods for Intrusion Detection Systems: A Survey. *Appl. Sci.* **2019**, *9*, 4396. [CrossRef]
8. Sarker, I.H.; Kayes, A.; Badsha, S.; Alqahtani, H.; Watters, P.; Ng, A. Cybersecurity data science: An overview from machine learning perspective. *J. Big Data* **2020**, *7*, 1–29. [CrossRef]
9. Hindy, H.; Brosset, D.; Bayne, E.; Seeam, A.K.; Tachtatzis, C.; Atkinson, R.; Bellekens, X. A Taxonomy of Network Threats and the Effect of Current Datasets on Intrusion Detection Systems. *IEEE Access* **2020**, *8*, 104650–104675. [CrossRef]
10. Canadian Institute for Cybersecurity. 2017. Intrusion Detection Evaluation Dataset (CICIDS2017). Available online: http://www.unb.ca/cic/datasets/ids-2017.html (accessed on 7 October 2019).
11. Panigrahi, R.; Borah, S. A detailed analysis of CICIDS2017 dataset for designing Intrusion Detection Systems. *Int. J. Eng. Technol.* **2018**, *7*, 479–482.
12. Vinayakumar, R.; Alazab, M.; Soman, K.P.; Poornachandran, P.; Al-Nemrat, A.; Venkatraman, S. Deep Learning Approach for Intelligent Intrusion Detection System. *IEEE Access* **2019**, *7*, 41525–41550. [CrossRef]
13. Hossain, M.D.; Ochiai, H.; Doudou, F.; Kadobayashi, Y. SSH and FTP brute-force Attacks Detection in Computer Networks: LSTM and Machine Learning Approaches. In Proceedings of the 2020 5th International Conference on Computer and Communication Systems (ICCCS), Shanghai, China, 22–24 February 2020; pp. 491–497.
14. Yulianto, A.; Sukarno, P.; Suwastika, N.A. Improving adaboost-based intrusion detection system (IDS) performance on CIC IDS 2017 dataset. *J. Phys. Conf. Ser.* **2019**, *1192*, 012018. [CrossRef]
15. Aksu, D.; Ali Aydin, M. Detecting Port Scan Attempts with Comparative Analysis of Deep Learning and Support Vector Machine Algorithms. In Proceedings of the 2018 International Congress on Big Data, Deep Learning and Fighting Cyber Terrorism (IBIGDELFT), Ankara, Turkey, 3–4 December 2018; pp. 77–80.
16. Abdulrahman, A.A.; Ibrahem, M.K. Evaluation of DDoS attacks Detection in a New Intrusion Dataset Based on Classification Algorithms. *Iraqi J. Inf. Commun. Technol.* **2018**, *1*, 49–55. [CrossRef]
17. Bishop, C.M. *Pattern Recognition and Machine Learning*; Springer: New York, NY, USA, 2006.
18. Bolón-Canedo, V.; Alonso-Betanzos, A. Ensembles for Feature Selection: A Review and Future Trends. *Infor. Fusion* **2019**, *52*, 1–12. [CrossRef]
19. Hamon, J. Combinatorial Optimization for Variable Selection in High Dimensional Regression: Application in Animal Genetic. Ph.D. Thesis, Université des Sciences et Technologie de Lille, Lille I, France, 2013.
20. Phuong, T.M.; Lin, Z.; Altman, R.B. Choosing SNPs using feature selection. In Proceedings of the 2005 IEEE Computational Systems Bioinformatics Conference (CSB'05), Stanford, CA, USA, 8–11 August 2005; pp. 301–309.
21. Hernandez, J.C.H.; Duval, B.; Hao, J.K. A genetic embedded approach for gene selection and classification of microarray data. In *European Conference on Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics*; Springer: Berlin/Heidelberg, Germany, 2007; pp. 90–101.
22. Law, M.H.C.; Figueiredo, M.A.T.; Jain, A.K. Simultaneous feature selection and clustering using mixture models. *IEEE Trans. Pattern Anal. Mach. Intell.* **2004**, *26*, 1154–1166. [CrossRef] [PubMed]
23. Pudil, P.; Novovičová, J.; Choakjarernwanit, N.; Kittler, J. Feature selection based on the approximation of class densities by finite mixtures of special type. *Pattern Recogn.* **1995**, *28*, 1389–1398. [CrossRef]
24. Mitra, P.; Murthy, C.A.; Pal, S.K. Unsupervised feature selection using feature similarity. *IEEE Trans. Pattern Anal. Mach. Intell.* **2002**, *24*, 301–312. [CrossRef]
25. McCulloch, W.S.; Pitts, W. A logical calculus of the ideas immanent in nervous activity. *Bull. Math. Biophys.* **1943**, *5*, 115–133. [CrossRef]
26. McClelland, J.L.; Rumelhart, D.E. A distributed model of human learning and memory. In *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 2: Psychological and Biological Models*; MIT Press: Cambridge, MA, USA, 1986; pp. 170–215.

27. Xu, G.; Qiu, S.; Ahmad, H.; Xu, G.; Guo, Y.; Zhang, M.; Xu, H. A Multi-Server Two-Factor Authentication Scheme with Un-Traceability Using Elliptic Curve Cryptography. *Sensors* **2018**, *18*, 2394. [CrossRef] [PubMed]
28. Qiu, S.; Wang, D.; Xu, G.; Kumari, S. Practical and Provably Secure Three-Factor Authentication Protocol Based on Extended Chaotic-Maps for Mobile Lightweight Devices. *IEEE Trans. Depend. Secure Comput.* **2020**. [CrossRef]
29. Kuhn, M.; Johnson, K. *Recursive Feature Elimination | Feature Engineering and Selection: A Practical Approach for Predictive Models*; Taylor & Francis Group: Abingdon, UK, 2019; Chapter 11.3.
30. Services, O. SlowHTTPTest | Penetration Testing Tools. 2020. Available online: https://tools.kali.org/stress-testing/slowhttptest (accessed on 29 July 2020).
31. Hindy, H.; Atkinson, R.; Tachtatzis, C.; Colin, J.N.; Bayne, E.; Bellekens, X. Utilising Deep Learning Techniques for Effective Zero-Day Attack Detection. *Electronics* **2020**, *9*, 1684. [CrossRef]