Journal of Cybersecurity and Privacy

MDPI

Article

MOCA: A Network Intrusion Monitoring and Classification System

Jessil Fuhr [1,*], Feng Wang [1] and Yongning Tang [2]

[1] School of Engineering, Liberty University, Lynchburg, VA 24502, USA
[2] School of Information Technology, Illinois State University, Normal, IL 61761, USA
* Correspondence: jbarboza@liberty.edu

Abstract: Optimizing the monitoring of network traffic features to detect abnormal traffic is critical. We propose a two-stage monitoring and classification (MOCA) system requiring fewer features to detect and classify malicious network attacks. The first stage monitors abnormal traffic, and the anomalous traffic is forwarded for processing in the second stage. A small subset of features trains both classifiers. We demonstrate MOCA's effectiveness in identifying attacks in the CICIDS2017 dataset with an accuracy of 99.84% and in the CICDDOS2019 dataset with an accuracy of 93%, which significantly outperforms previous methods. We also found that MOCA can use a pre-trained classifier with one feature to distinguish DDoS and Botnet attacks from normal traffic in four different datasets. Our measurements show that MOCA can distinguish DDoS attacks from normal traffic in the CICDDOS2019 dataset with an accuracy of 96% and DDoS attacks in non-IoT and IoT traffic with an accuracy of 99.94%. The results emphasize the importance of using connection features to discriminate new DDoS and Bot attacks from benign traffic, especially with insufficient training samples.

Keywords: intrusion detection systems; feature selection; DDoS attacks

check for updates

Citation: Fuhr, J.; Wang, F.; Tang, Y. MOCA: A Network Intrusion Monitoring and Classification System. *J. Cybersecur. Priv.* **2022**, *2*, 629–639. https://doi.org/10.3390/jcp2030032

Academic Editor: Steven Furnell

Received: 11 April 2022
Accepted: 10 August 2022
Published: 15 August 2022

1. Introduction

Feature selection is a critical process in network intrusion detection. Many previous works, such as [1,2], have shown that the accuracy of machine learning-based or deep learning-based methods is heavily affected by the feature space. The main challenge in designing efficient ML/DL-based intrusion detection is to choose a subset of relevant features without negatively affecting classification accuracy. Previous work [3] has shown that different features contribute differently to detecting attack classes. Many efforts have concentrated on selecting important features for a monolithic classifier to detect and classify malicious attacks. For example, a single ML/DL-based classifier is proposed in [4,5]. In previous work [6], a Random Forest model trained by 22 selected features of the CICIDS2017 [7] dataset achieved an accuracy of 99.86%. Although the number of features is reduced, it is still not practical to monitor and log all selected features for detecting anomalous traffic and attacks, especially when coping with a large amount of traffic. Previous work [2] suggests ten features from the same dataset, but it cannot identify some attacks with reasonable accuracy. Therefore, it is challenging to design a lightweight intrusion detection system that efficiently monitors fewer features to detect abnormal traffic and attacks.

In this paper, we propose a two-stage MOnitoring and ClAssification (MOCA) system to detect and classify malicious network attacks. In the first phase of MOCA, a binary classifier monitors abnormal traffic. The monitor is trained by a small subset of features, which are carefully selected to distinguish anomalous traffic and attacks from normal traffic. The anomalous traffic captured by the monitor is forwarded for processing in a multi-class classifier at the second stage, where the attack classes are detected. With fewer features, MOCA makes it possible to develop a lightweight intrusion detection system to keep track

of flow statistics. When flow exhibits behavior outside its normal range, it is monitored further at the second stage. Compared with related work, a two-step hybrid method that contains several binary classifiers and one aggregation module (KNN) is presented in [8]. MOCA contains one single binary classifier and one multi-class classifier that identifies attack classes, meaning MOCA is more lightweight than the work presented in [8].

The effectiveness of MOCA was evaluated by using the CICIDS2017 dataset [7] and the CICDDOS2019 dataset [9]. Our measurement results show that MOCA can effectively detect anomalous traffic and attacks with a few features. More specifically, using three to five features at the first stage, MOCA successfully differentiates anomalous traffic from normal traffic with an accuracy of almost 100%. Those features are directly derived from packet-level data or flow-level data, which makes MOCA suitable for monitoring real-time traffic by using sketching-based methods, such as count-min sketch [10,11].

Using 10 features overall, MOCA can identify attack classes in the CICIDS2017 dataset with an accuracy of 99.84% and attack classes in the CICDDOS2019 dataset with an accuracy of 93%, which significantly outperforms previous methods [2,6]. Using as few as eight features, MOCA has a much higher accuracy rate (i.e., 98%) on the CICIDS2017 dataset and 91% on the CICDDOS2019 dataset than when using only six features. Our measurement results demonstrate that connection features are highly useful for intrusion detection, and MOCA possesses the highest accuracy rate among the other methods designed for imbalanced attack classes.

Furthermore, MOCA mainly chooses *connection* features to design the first stage classifier. Our measurement results show that benign traffic and DDoS attacks are the two classes benefiting from connection features. The CICIDS2017 dataset trained four decision tree-based binary classifiers to identify DDoS attacks. Each classifier is trained by one of the connection features, making it a lightweight classifier. Then, the pre-trained classifier is used to distinguish DDoS and Bot attacks from normal traffic in the ToN-IoT [12], BoT-IoT [12], CICIDS2018 [7], and CICDDOS2019 datasets. Note that these datasets contain traditional IT and IoT traffic with different DDoS and Botnet attacks. We found that pre-trained classifiers can distinguish most DDoS and Botnet attacks from normal traffic. For example, the classifier can distinguish DDoS attacks from normal traffic in the CICDDOS2019 dataset with an accuracy of 96% and DDoS attacks in non-IoT and IoT traffic with an accuracy of 99.94%. To the best of our knowledge, MOCA is the first intrusion detection system to efficiently use a pre-trained model without retraining. Our measurements show that connection features [13] effectively detect new DDoS and Bot attacks, especially when there are not enough training samples. They are also efficient features for minimizing the training and execution time of MOCA.

The rest of the paper is organized as follows. In Section 3, connection features are introduced. In Section 4, the proposed detection system is presented. The performance of our detection system is evaluated in Section 5. Some related works are explored in Section 2, and the paper concludes with a summary in Section 6.

2. Related Works

Almost all previous work has focused on selecting important features without distinguishing connection features [5,14–17]. For example, the results from previous works based on KDD-99 [18–20], NSL-KDD [21,22], and UNSW-NB15 [23] show that connection features typically are not listed as top-10 important features [5,16,17]. An SVM-based attack detection system is proposed in [24]. A modular deep neural network is proposed in [25]. An analysis of the importance of CICIDS2017 features that used permutation importance to reduce the required features from the original 69 to 10 is reported in [2]. The work presented in [8] proposes a two-step hybrid method based on binary classification and kNN. It contains several binary classifiers and one aggregation module to detect attack classes effectively.

3. Connection Features

Researchers classify features into two particular classes: *basic features* and *connection features*. Basic features represent a single connection, while connection features are extracted from multiple connections. The preeminent objective of the two features is to identify two general types of attacks in network intrusion detection: attacks that require single connections, such as Web SQL Injection, and attacks that contain multiple connections, such as DDoS and Botnet attacks.

Distributed Denial of Service (DDoS) is a malicious attack in which numerous systems use malicious traffic to confuse a target server. In the CICDDoS2019[9] dataset, there are 13 DDoS attacks, such as DNS, MSSQL, SSDP, and UDPLag attacks. The following connection features are used to find the behavior of a client with a 100-connection window:

- When there are multiple connections whose source IP address is the same as the current connection, the *source count* is extracted.
- When there are multiple connections whose destination IP address is the same as the current connection, the *dst count* is extracted.
- When there are multiple connections whose source port is the same as the current connection, the *sport count* is extracted.
- When there are multiple connections whose destination port is the same as that of the current connection, the *dport count* is extracted.
- When there are multiple connections whose destination and source IP addresses are the same as those of the current connection, the *dst source count* is extracted.
- When there are multiple connections whose destination IP address and source port are the same as those of the current connection, the *dst sport count* is extracted.
- When there are multiple connections whose source IP address and destination port are the same as those of the current connection, the *source dport count* is extracted.

The datasets used in this paper do not contain connection features. As a result, based on the original files, the seven connection features were derived for each sample as mentioned above.

4. MOCA Design

MOCA consists of a two-step approach: (1) monitor and detect anomalies, and (2) classify attacks. Figure 1 depicts the framework of MOCA. Instead of training one ML/DL-based classifier, MOCA trains a binary classifier and a multi-class classifier. The binary classifier is to distinguish abnormal traffic from normal traffic. It is necessary to mention that all traffic is available prior to feature calculations. The multi-class classifier performs multi-class classification on the abnormal traffic to identify attack classes.

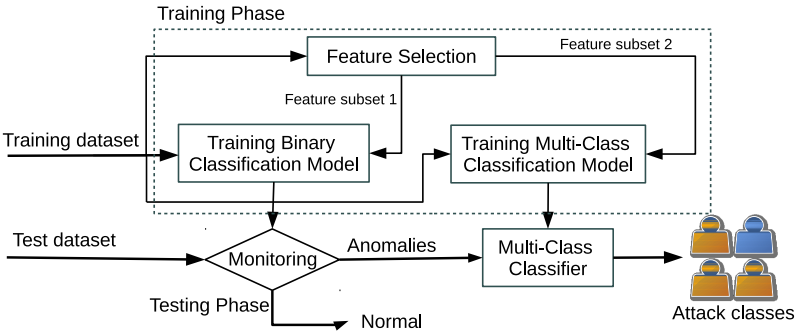


Figure 1. Framework of MOCA.

Two separated feature subsets are used for training classifiers. First, a small set of features is selected for a binary classifier to identify abnormal traffic. The second feature subset is selected for a multi-class classifier to identify attack classes. Various feature selection methods can determine the two feature subsets. The rest of this section focuses on the feature selection methods.

4.1. Feature Importance

This study measures feature importance using the following algorithms: information gain, correlation coefficients, and SHAP (SHapley Additive exPlanations) values. Consider S a set of training-set samples with corresponding labels. Assuming there are m classes, the training set contains c_i samples of class C_i, and N is the total number of samples in the training set. The information needed to classify a given sample is calculated by the following equation:

$$I(c_i, c_2, \ldots, c_m) = -\sum_{i=1}^{m} \frac{c_i}{N} \log_2(\frac{c_i}{N}) \tag{1}$$

Feature F with values $\{f_1, f_2, \ldots, f_v\}$ can divide the training set into v subsets $\{S_1, S_2, \ldots, S_v\}$, where S_j is the subset with value f_j for feature F. Furthermore, let S_j contain s_{ij} samples of class i. Entropy of feature F is:

$$E(F) = \sum_{j=1}^{v} \frac{s_{1j} + \ldots + s_{mj}}{N} \times I(s_{1j}, \ldots, s_{mj}) \tag{2}$$

Information gain for F can be calculated as:

$$Gain(F) = I(s_i, s_2, \ldots, s_m) - E(F) \tag{3}$$

It is noticeable that the information metric is a good measurement among different feature selection metrics to quantify the importance of features.

Correlation coefficients evaluate a subset of highly correlated features within an attack class but not correlated with each other. The correlation coefficient of a feature subset F containing k features is given by [5]:

$$Merit_s = \frac{k \times \overline{r_{cf}}}{\sqrt{k + k \times (k-1) \times \overline{r_{ff}}}}$$

where $\overline{r_{cf}}$ is the mean feature-class correlation, and $\overline{r_{ff}}$ is the average feature inter-correlation

The SHAP value [26] of a feature is it contribution to the payout, it is weighted and summed over all possible feature combinations:

$$\phi_j(val) = \sum_{S \subseteq X \setminus x_j} \frac{|S|!(p - |S| - 1)!}{p!} (val(S \cup x_j)) - val(S)) \tag{4}$$

where S is a subset of all features used in the model, $X = \{x_1, \ldots, x_p\}$ describes the vector of feature values of the instance to be explained, and p is the number of features; $val_x(S)$ is the prediction of feature values in set S that are marginalized over features not included in set S.

4.2. Feature Selection

Feature selection techniques are divided into filter-based, wrapper-based, and embedded-based. In filter-based methods, someone selects features involving no ML algorithm. Filter methods are useful with respect to computing time. Wrapper-based methods employ a supervised learning algorithm to choose feature subsets. The two fundamental wrapper-based approaches adopted are Step Forward feature Selection (SFS) and Sequential Backward Selection (SBS). Embedded techniques incorporate optimal feature selection into an ML-based classifier.

In this paper, the integration of filter and embedded methods was necessary to select potential features, called *pre-selected features*, to improve the effectiveness of searching for the optimal feature subsets. Information gain and correlation coefficient were used as filter-based methods. Decision Tree, Random Forest, and XGBoost were used as embedded

methods to calculate the importance of the pre-selected features. From each approach, the top-20 important features were derived. Then, a majority vote approach was taken to derive the pre-selected features.

After feature pre-selection, a feature-searching Algorithm 1 was proposed to select optimal feature subsets (f_1 and f_2) for the two stages of MOCA. In this algorithm, pre-selected features are used to train a binary classifier. A SHAP TreeExplainer is used to derive important features of the model. Features with top-20 SHAP values are used as the starting features for the SBS method to find the minimum set of features. Each feature is deleted one at a time; classification accuracy is computed for all subsets with the remaining features, and the worst feature is discarded. To train the optimum model, cross-validation and hyper-parameter optimization played a role in bias reduction.

Algorithm 1 Selecting (k_1, k_2) features in MOCA.

Input: pre-selected features f_{pre}, training dataset D_{train}, two classifiers m_1 and m_2, and the number of features (k_1, k_2)
Output: trained m_1 and m_2 with feature subsets f_1, f_2

1. Train a model m using D_{train} and f_{pre}
2. Explain m using shap.treeExplainer()
3. Derive top-20 features f_{20} according to SHAP values $f_1 = f_2 = f_{20}$
4. Train m_1 using D_{train} and f_1
5. Train m_2 using D_{train} and f_2
6. Search for the worst feature x^- using $SBC(m_1)$ and $SBC(m_2)$
7. Update $f_1 = f_1 - x^-$ and $f_2 = f_2 - x^-$
8. **if** $|f_1| == k_1$ and $|f_2| == k_2$ **then**
9. | **return** m_1, m_2, f_1, and f_2;
10. **end**
11. Go to 5;

4.3. Binary and Multi-Class Classifiers

This paper uses Decision Tree, Random Forest, XGBoost, and multilayer perceptron (MLP) classifiers as binary and multi-class classifiers. These models are used to investigate the effect of selected features on the classification accuracy of classifiers. They minimize a feature subset that maximizes the binary classification performance while minimizing the number of selected features.

4.4. Performance Metrics

To effectively evaluate the performance of MOCA, four performance metrics, accuracy, precision, recall, and F1-score, were critical. More specifically, the following four metrics are considered for evaluating the performance of intrusion detection methods:

- Accuracy: the sum of flows classified correctly with respect to the total number of flows. It is written by:

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

  where TP is the number of true positives or simply the true positives, TN is the true negatives, FP is the false positives, and FN is the false negatives.
- Precision: positive predictive value, equal to:

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

- Recall: sensitivity or Detection Rate (DR), which is equal to:

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

- F1-Score: the harmonic mean of the Precision and Recall. In other words, it is a statistical technique for examining the system's accuracy by considering both precision and recall:

$$\text{F1}_{\text{score}} = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$$

5. Measurement Results

In this section, the performance of MOCA in classifying attack classes was evaluated in the CICIDS2017 and CICDDOS2019 datasets. A total of 84 features in CICIDS2017 and 88 features in CICDDOS2019, including labels, were extracted based on real-world data (PCAPs). Flow ID, timestamp, IP, source, and destination port features were removed. In addition, 8 extra features in CICIDS2017 and 13 extra features in CICDDOS2019 were removed because they were all zeros. Based on the original features, connection features were derived for each sample. As a result, the datasets were left with 68 basic features in CICIDS2017 and 67 basic features in CICDDOS2019.

5.1. Binary Classification Accuracy

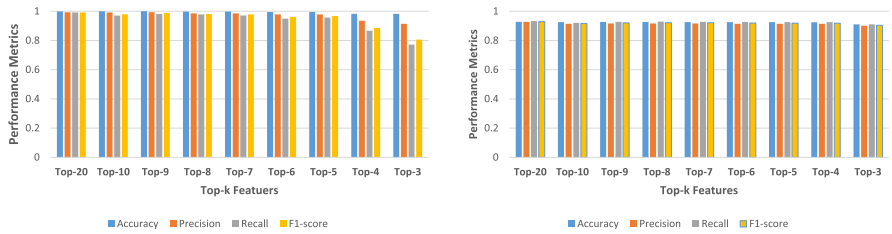
Table 1 shows the classification accuracy of an XGBoost-based classifier trained on different numbers of features. The table shows that using top-10 important features, including three connection features, can detect almost all abnormal samples in both datasets. Combining top-2 basic features ("Bwd Packet Length Min", "Subflow Fwd Bytes") with top-3 connection features (*dport_count*, *dst_src_count*, *src_dport_count*) can distinguish abnormal traffic from normal traffic with an accuracy of 98.9% in the CICIDS2017 dataset and 99.98% in the CICDDOS2019 dataset. Using the three connection features detects 95% of abnormal cases in the CICIDS2017 dataset and 99.91% of attacks in the CICDDOS2019 dataset. MOCA has a higher accuracy rate for the CICDDOS2019 dataset. All attacks in the CICDDOS2019 dataset are DDoS attacks, which are sensitive to connection features. One aspect to highlight regarding the importance of connection features is that using top-5 basic features cannot detect all attacks.

Table 1. Performance of binary classifier trained with different selected features.

	CICIDS2017				CICDDOS2019			
Selected Features	**Accuracy**	**Prec**	**Rec**	**F1-Score**	**Accuracy**	**Prec**	**Rec**	**F1-Score**
Top-20 F w/CF	**1.000000**	0.999952	0.999952	0.999952	**1.000000**	0.999987	0.999987	0.999987
Top-10 F w/CF	0.996000	0.995979	0.995978	0.995978	**1.000000**	**1.000000**	**1.000000**	**1.000000**
Top-5 F w/ 3 CF	0.989400	0.989414	0.989368	0.989361	0.999800	0.999796	0.999796	0.999796
Top-3 CF	0.951900	0.951822	0.951863	0.951871	0.999100	0.999060	0.999055	0.999056
Top-20 BF	0.999800	0.999846	0.999846	0.999846	**1.000000**	0.999974	0.999974	0.999974
Top-10 BF	0.991200	0.991227	0.991209	0.991211	**1.000000**	0.999883	0.999883	0.999883
Top-5 BF	0.970700	0.971417	0.970745	0.970795	0.986600	0.986546	0.986613	0.986345
Top-3 BF	0.915100	0.918395	0.915075	0.914359	0.982900	0.982693	0.982857	0.982448

5.2. Multi-Class Classification Accuracy

Decision Tree (DT), Random Forest (RF), and XGBoost are used as multi-class classifiers. This paper presents the measurement results from an XGBoost model because it has better performance than other models. Figure 2 shows the performance of the classifier trained by top-k selected features. The top-k features are selected by Algorithm 1. The classifier trained using top-5 and top-3 features from both datasets had a very high detection rate.



(**a**) (**b**)

Figure 2. Performance of multi-class classifier trained by top-k selected features: (**a**) CICIDS2017; (**b**) CICDDOS2019.

The classification results for 14 classes in the CICIDS2017 dataset are shown in Table 2. The last three entries in Table 2 are the three attacks involving multiple connections. The results clearly show that MOCA can detect all three attacks, which is better than previous work [2]. In [2], a Random Forest model was trained by ten basic features. We found that MOCA can detect the majority of imbalanced web attacks. On the contrary, the model in [2] could only detect a few attacks.

Table 2. Comparison of results of MOCA vs. previous work [2] based on CICIDS2017 dataset.

	MOCA (Top-20)			**MOCA (Top-7)**			**MOCA (Top-5)**			**RF (10) [2]**		
Class	**Prec**	**Rec**	**F1**	**Prec**	**Rec**	**F1**	**Prec**	**Rec**	**F1**	**Prec**	**Rec**	**F1**
DoS GoldenEye	**1.00**	**1.00**	**1.00**	0.99	**1.00**	0.99	0.99	1.00	0.99	**1.00**	0.99	0.99
DoS Hulk	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.99	**1.00**	0.99	0.99	**1.00**	0.99
DoS Slowhttptest	**1.00**	**1.00**	**1.00**	0.97	0.96	0.97	0.86	0.85	0.86	0.83	0.83	0.83
DoS slowloris	**1.00**	**1.00**	**1.00**	0.98	0.98	0.98	0.93	0.93	0.93	0.98	0.98	0.98
FTP-Patator	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.99	0.99
Heartbleed	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
PortScan	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.99	**1.00**	0.99
SSH-Patator	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.99	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
(Brute Force)	0.97	0.97	0.97	0.95	0.91	0.93	0.94	0.90	0.92	0.58	0.45	0.52
(Sql Inj)	**1.00**	**1.00**	**1.00**	**1.00**	0.86	0.92	**1.00**	0.86	0.92	**1.00**	0.80	0.89
Web Attack XSS	0.93	0.92	0.93	0.92	0.89	0.91	0.99	0.87	0.93	0.68	0.76	0.72
Bot	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.83	0.79	0.81
DDoS	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.99	0.99
Infiltration	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.88	0.88

The results shows that MOCA improves performance over other methods, especially for imbalanced attack classes. Most previous works [27] showed poor performance in detecting the web attacks in the CICIDS2017 dataset due to the imbalanced data based on the nature of different web attack behaviors. The SMOTE method [28] has been applied to address the issue of imbalance by oversampling the imbalanced samples in the training dataset. MOCA presents very high detection accuracy in all three web attack classes without using the SMOTE method.

Table 3 shows the performance of MOCA in detecting 13 classes in the CICDDOS2019 dataset. Similarly, using three connection features to train the multi-class classifier can identify more attack classes than can a Random Forest model trained with all basic features.

Table 3. Comparison of results for detection accuracy of MOCA vs. Random Forest model based on CICDDoS2019 dataset.

	MOCA (Top-20)			MOCA (Top-7)			MOCA (Top-3)			Random Forest		
Classes	**Prec**	**Rec**	**F1**	**Prec**	**Rec**	**F1**	**Prec**	**Rec**	**F1**	**Prec**	**Rec**	**F1**
DNS	**1.00**	**1.00**	**1.00**	0.99	0.98	0.99	0.97	0.97	0.97	0.99	0.99	0.99
MSSQL	0.99	0.98	0.98	0.98	0.98	0.98	0.98	0.97	0.98	0.98	0.97	0.98
NTP	**1.00**	**1.00**	**1.00**	0.99	0.99	0.99	0.97	0.97	0.97	0.99	0.99	0.99
SNMP	0.94	0.92	0.93	0.93	0.92	0.92	0.82	0.91	0.86	0.92	0.91	0.91
SSDP	0.85	0.73	0.79	0.84	0.70	0.76	0.83	0.70	0.76	0.78	0.67	0.72
LDAP	0.95	0.99	0.97	0.95	0.99	0.97	0.94	0.85	0.90	0.95	0.99	0.97
NetBIOS	0.90	0.92	0.90	0.91	0.91	0.88	0.91	0.88	0.89	0.88	0.89	0.89
Portmap	0.90	0.91	0.91	0.87	0.92	0.89	0.87	0.93	0.90	0.88	0.88	0.88
Syn	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
TFTP	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
UDP	0.85	0.95	0.90	0.85	0.96	0.90	0.85	0.96	0.90	0.82	0.93	0.87
UDP-lag	0.84	0.82	0.83	0.84	0.82	0.83	0.83	0.82	0.83	0.78	0.75	0.77
WebDDos	0.92	**1.00**	0.96	0.78	0.93	0.85	0.75	0.89	0.82	0.88	0.96	0.92

5.3. Effectiveness of Pre-Trained Binary Classifier

To further investigate the benefit of using connection features, four Decision Tree-based binary classifiers were designed. Each classifier was trained by one feature (*dst_src_count*, *src_count*, *dst_count*, or *src_dport*) and DDoS attacks from the CICIDS2017 dataset. Then, DDoS and Bot attacks were used from ToN-IoT [29], BoT-IoT [12], CICIDS2018 [7], and CICDDOS2019 datasets to evaluate the pre-trained classifiers without retraining. Pre-trained decision trees distinguished DDoS and Botnet attacks from normal traffic. Note that these datasets contain traditional IT and IoT traffic (in BoT-IoT and ToN-IoT datasets) with different types of DDoS attacks. Each pre-trained binary classifier is a lightweight decision tree trained by one feature. The result is shown in Table 4. The table shows that pre-trained classifiers can distinguish most DDoS and Botnet attacks from normal traffic. For example, in the CICDDOS2019 dataset, they had an accuracy of 96%. For the BoT-IoT dataset, a pre-trained classifier detected almost all DDoS attacks in non-IoT and IoT traffic with 99.94% accuracy.

We noticed that pre-trained classifiers contribute differently to detecting attack classes. For example, *dst_src_count* was used for the CICDDOS2019 dataset, and *src_dport* was used for the BoT-IoT dataset. The other two features, *dst_count* and *src_count*, were used to classify DDoS attacks in CICIDS2018 and ToN-IoT datasets, respectively. The results show that connection features can be selected as the potential features for training a binary classifier instead of searching all features. The results also show that pre-trained binary classifiers can help detect new DDoS and Botnet attacks when training samples are unavailable. In addition, accuracy can be further improved by using the selected connection features and one or two basic features from the new dataset to train the classifier. Therefore, MOCA can minimize training time.

Table 4. Performance of binary classifiers trained by CICIDS2017 dataset and tested on four other datasets.

Dataset	Accuracy	Class	Precision	Recall	F1
ToN-IoT [29]	0.8804	Benign	0.97	0.69	0.80
		Attack	0.85	0.99	0.91
BoT-IoT [12]	0.9994	Benign	0.97	0.96	0.96
		Attack	**1.00**	**1.00**	**1.00**
CICIDS2018 [7]	0.9482	Benign	0.90	**1.00**	0.95
		Attack	**1.00**	0.90	0.95
CICDDOS2019 [9]	0.9667	Benign	0.99	0.70	0.82
		Attack	0.96	**1.00**	0.98

5.4. Comparing MOCA with Previous Works

Table 5 compares the results of MOCA with previous works. Because previous works used recall or decision rate, it was necessary to only compare their recall values with ours. From the table, the values for MOCA are higher than previous approaches. For example, SGM only achieved a slightly high recall value in Web Attack XSS. This observation implies that MOCA can efficiently address the effects of imbalanced malicious attacks on classification accuracy.

Table 5. Performance comparison of our method vs. previous works based on CICIDS2017 dataset.

Class	MOCA Method	SGM [30]	UDBB [31]	WiSARD [32]	LSTM [33]
Benign	**1.00**	**1.00**	**1.00**	0.97	0.87
Bot	**1.00**	**1.00**	**1.00**	0.14	0.83
DDoS	**1.00**	**1.00**	**1.00**	0.54	0.71
DoS GoldenEye	**1.00**	**1.00**	**1.00**	0.48	0.74
DoS Hulk	**1.00**	**1.00**	0.99	0.67	0.74
DoS Slowhttptest	**1.00**	**1.00**	0.99	0.23	0.71
DoS slowloris	**1.00**	**1.00**	0.99	0.79	0.71
FTP-Patator	**1.00**	**1.00**	**1.00**	0.00	0.91
Heartbleed	**1.00**	**1.00**	**1.00**	0.80	0.96
Infiltration	**1.00**	**1.00**	**1.00**	0.50	0.95
PortScan	**1.00**	**1.00**	**1.00**	0.51	0.98
SSH-Patator	**1.00**	**1.00**	**1.00**	0.00	0.87
Web Attack Brute Force	0.97	0.96	0.94	0.47	0.71
Web Attack Sql Injection	**1.00**	**1.00**	**1.00**	0.00	0.71
Web Attack XSS	0.92	0.93	0.88	0.12	0.71

6. Conclusions

In this paper, we designed a two-stage monitoring and classification system. Our measurements confirm that MOCA significantly outperforms previous methods. MOCA can distinguish DDoS and Bot attacks from normal and IoT traffic with high accuracy using a pre-trained model. After pre-training by the CICIDS2017 dataset, MOCA could distinguish DDoS attacks from normal traffic in the CICDDOS2019 dataset with an accuracy of 96% and DDoS attacks in non-IoT and IoT traffic with an accuracy of 99.94%. Our work shows that connection features are valuable for detecting new DDoS and Bot attacks, especially when training datasets are unavailable. They are efficient features for minimizing the training and execution time of MOCA.

Author Contributions: Conceptualization, F.W. and Y.T.; methodology, F.W., Y.T., J.F.; development; F.W., Y.T., J.F. software, F.W., Y.T., J.F.; validation, F.W., Y.T., J.F.; formal analysis, F.W., Y.T., J.F.; investigation, F.W., Y.T., J.F.; resources, F.W., Y.T., J.F.; data curation, F.W., Y.T., J.F.; writing—original draft preparation, F.W.; writing—review and editing, F.W., Y.T., J.F.; visualization, F.W., Y.T., J.F.; supervision, F.W.; project administration, F.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

1. Kasongo, S.M.; Sun, Y. A Deep Learning Method With Filter Based Feature Engineering for Wireless Intrusion Detection System. *IEEE Access* **2019**, *7*, 38597–38607. [CrossRef]
2. Reis, B.; Maia, E.; Praça, I. Selection and Performance Analysis of CICIDS2017 Features Importance. In *Foundations and Practice of Security—12th International Symposium, FPS 2019*; Lecture Notes in Computer Science; Benzekri, A., Barbeau, M., Gong, G., Laborde, R., García-Alfaro, J., Eds.; Springer: Berlin/Heidelberg, Germany, 2019, Volume 12056, pp. 56–71.
3. Lian, W.; Nie, G.; Jia, B.; Shi, D.; Fan, Q.; Liang, Y. An Intrusion Detection Method Based on Decision Tree-Recursive Feature Elimination in Ensemble Learning. *Math. Probl. Eng.* **2020**, *2020*, 2835023. [CrossRef]
4. Ambusaidi, M.A.; He, X.; Nanda, P.; Tan, Z. Building an Intrusion Detection System Using a Filter-Based Feature Selection Algorithm. *IEEE Trans. Comput.* **2016**, *65*, 2986–2998. [CrossRef]
5. Injadat, M.; Moubayed, A.; Nassif, A.B.; Shami, A. Multi-Stage Optimized Machine Learning Framework for Network Intrusion Detection. *IEEE Trans. Netw. Serv. Manag.* **2020**, *18*, 1803–1816. [CrossRef]
6. Kurniabudi.; Stiawan, D.; Darmawijoyo.; Idris, M.Y.B.; Bamhdi, A.M.; Budiarto, R. CICIDS-2017 Dataset Feature Analysis With Information Gain for Anomaly Detection. *IEEE Access* **2020**, *8*, 132911–132921. [CrossRef]
7. Sharafaldin, I.; Lashkari, A.H.; Ghorbani, A.A. Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization. In *ICISSP 2018*; Mori, P., Furnell, S., Camp, O., Eds.; SciTePress: Setubal, Portugal, 2018; pp. 108–116.
8. Li, L.; Yu, Y.; Bai, S.; Hou, Y.; Chen, X. An Effective Two-Step Intrusion Detection Approach Based on Binary Classification and k -NN. *IEEE Access* **2018**, *6*, 12060–12073. [CrossRef]
9. Sharafaldin, I.; Lashkari, A.H.; Hakak, S.; Ghorbani, A.A. Developing Realistic Distributed Denial of Service (DDoS) Attack Dataset and Taxonomy. In Proceedings of the 2019 International Carnahan Conference on Security Technology (ICCST), Chennai, India, 1–3 October 2019; pp. 1–8.
10. Cormode, G.; Muthukrishnan, S. An improved data stream summary: the count-min sketch and its applications. *J. Algorithms* **2005**, *55*, 58–75. [CrossRef]
11. Wang, F.; Gao, L. Simple and Efficient Identification of Heavy Hitters Based on Bitcount. In Proceedings of the 20th IEEE International Conference on High Performance Switching and Routing, HPSR 2019, Xi'an, China, 26–29 May 2019; pp.1–6.
12. Koroniotis, N.; Moustafa, N.; Sitnikova, E.; Turnbull, B.P. 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]
13. Fuhr, J.; Hanna, I.; Wang, F.; Tang, Y. The Diminished Importance of Connection-based Features in Intrusion Detection. In Proceedings of the 40th IEEE International Performance Computing and Communications Conference, Austin, TX, USA, 29–31 October 2021; p. 10.
14. Javidi, M.M.; Mansouri, S. Intrusion detection system using an ant colony gene selection method based on information gain ratio using fuzzy rough sets. *AUT J. Model. Simul.* **2019**, *51*, 33–44.
15. Maza, S.; Touahria, M. Feature Selection Algorithms in Intrusion Detection System: A Survey. *KSII Trans. Int. Inf. Syst.* **2018**, *12*, 5079–5099.
16. Moustafa, N.; Slay, J. The Significant Features of the UNSW-NB15 and the KDD99 Data Sets for Network Intrusion Detection Systems. In Proceedings of the 4th International Workshop on Building Analysis Datasets and Gathering Experience Returns for Security, BADGERS@RAID 2015, Kyoto, Japan, 5 November 2015; pp. 25–31.
17. Binbusayyis, A.; Vaiyapuri, T. Identifying and Benchmarking Key Features for Cyber Intrusion Detection: An Ensemble Approach. *IEEE Access* **2019**, *7*, 106495–106513. [CrossRef]
18. Lee, W.; Stolfo, S.J.; Mok, K.W. Mining in a Data-Flow Environment: Experience in Network Intrusion Detection. In Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Diego, CA, USA, 15–18 August 1999; pp. 114–124.
19. Mohammadi, S.; Mirvaziri, H.; Ghazizadeh-Ahsaee, M.; Karimipour, H. Cyber intrusion detection by combined feature selection algorithm. *J. Inf. Secur. Appl.* **2019**, *44*, 80–88. [CrossRef]

20. Amiri, F.; Yousefi, M.R.; Lucas, C.; Shakery, A.; Yazdani, N. Mutual information-based feature selection for intrusion detection systems. *J. Netw. Comput. Appl.* **2011**, *34*, 1184–1199. [CrossRef]
21. Tavallaee, M.; Bagheri, E.; Lu, W.; Ghorbani, A.A. A detailed analysis of the KDD CUP 99 data set. In Proceedings of the 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications, CISDA 2009, Ottawa, ON, Canada, 8–10 July 2009; pp. 1–6.
22. Aljawarneh, S.; Aldwairi, M.; Yassein, M.B. Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model. *J. Comput. Sci.* **2018**, *25*, 152–160. [CrossRef]
23. Moustafa, N.; Slay, J. UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In Proceedings of the 2015 Military Communications and Information Systems Conference, MilCIS 2015, Canberra, Australia, 10–12 November 2015; pp. 1–6.
24. Jan, S.U.; Ahmed, S.; Shakhov, V.V.; Koo, I. Toward a Lightweight Intrusion Detection System for the Internet of Things. *IEEE Access* **2019**, *7*, 42450–42471. [CrossRef]
25. Leevy, J.L.; Khoshgoftaar, T.M. A survey and analysis of intrusion detection models based on CSE-CIC-IDS2018 Big Data. *J. Big Data* **2020**, *7*, 104. [CrossRef]
26. Shapley, L. *Quota Solutions op n-Person Games1*; Artin, E., Morse, M., Eds.; Princeton University Press: Princeton, NJ, USA, 1953; p. 343.
27. 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]
28. Cieslak, D.; Chawla, N.; Striegel, A. Combating imbalance in network intrusion datasets. In Proceedings of the 2006 IEEE International Conference on Granular Computing, Atlanta, GA, USA, 10–12 May 2006; pp. 732–737.
29. Moustafa, N.; Ahmed, M.; Ahmed, S. Data Analytics-enabled Intrusion Detection: Evaluations of ToN IoT Linux Datasets. *TrustCom* **2020**, *2020*, 727–735.
30. Zhang, H.; Huang, L.; Wu, C.Q.; Li, Z. An effective convolutional neural network based on SMOTE and Gaussian mixture model for intrusion detection in imbalanced dataset. *Comput. Netw.* **2020**, *177*, 107315. [CrossRef]
31. Abdulhammed, R.; Musafer, H.; Alessa, A.; Faezipour, M.; Abuzneid, A. Features Dimensionality Reduction Approaches for Machine Learning Based Network Intrusion Detection. *Electronics* **2019**, *8*, 322. [CrossRef]
32. De Gregorio, M.; Giordano, M. An experimental evaluation of weightless neural networks for multi-class classification. *Appl. Soft Comput.* **2018**, *72*, 338–354. [CrossRef]
33. Radford, B.J.; Richardson, B.D.; Davis, S.E. Sequence Aggregation Rules for Anomaly Detection in Computer Network Traffic. *arXiv* **2018**, arXiv:1805.03735.