Adoption of IP Truncation in a Privacy-Based Decision Tree Pruning Design: A Case Study in Network Intrusion Detection System

electronics

MDPI

Article

Adoption of IP Truncation in a Privacy-Based Decision Tree Pruning Design: A Case Study in Network Intrusion Detection System

Yee Jian Chew [1], Shih Yin Ooi [1,*], Kok-Seng Wong [2], Ying Han Pang [1] and Nicholas Lee [1]

[1] Faculty of Information Science and Technology, Multimedia University, Jalan Ayer Keroh Lama, Melaka 75450, Malaysia; chewyeejian@gmail.com (Y.J.C.); yhpang@mmu.edu.my (Y.H.P.); lmz.nicholas@gmail.com (N.L.)

[2] College of Engineering and Computer Science, VinUniversity, Vinhomes Ocean Park, Hanoi 100000, Vietnam; wong.ks@vinuni.edu.vn

* Correspondence: syooi@mmu.edu.my

Abstract: A decision tree is a transparent model where the rules are visible and can represent the logic of classification. However, this structure might allow attackers to infer confidential information if the rules carry some sensitive information. Thus, a tree pruning methodology based on an IP truncation anonymisation scheme is proposed in this paper to prune the real IP addresses. However, the possible drawback of carelessly designed tree pruning might degrade the performance of the original tree as some information is intentionally opted out for the tree's consideration. In this work, the 6-percent-GureKDDCup'99, full-version-GureKDDCup'99, UNSW-NB15, and CIDDS-001 datasets are used to evaluate the performance of the proposed pruning method. The results are also compared to the original unpruned tree model to observe its tolerance and trade-off. The tree model adopted in this work is the C4.5 tree. The findings from our empirical results are very encouraging and spell two main advantages: the sensitive IP addresses can be "pruned" (hidden) throughout the classification process to prevent any potential user profiling, and the number of nodes in the tree is tremendously reduced to make the rule interpretation possible while maintaining the classification accuracy.

Keywords: privacy-preserving; IP address truncation; C4.5 decision tree; pruning; network intrusion detection system

check for updates

Citation: Chew, Y.J.; Ooi, S.Y.; Wong, K.-S.; Pang, Y.H.; Lee, N. Adoption of IP Truncation in a Privacy-Based Decision Tree Pruning Design: A Case Study in Network Intrusion Detection System. *Electronics* **2022**, *11*, 805. https://doi.org/10.3390/electronics11050805

Academic Editors: Andrii Shalaginov and Mamoun Alazab

Received: 25 January 2022
Accepted: 1 March 2022
Published: 4 March 2022

1. Introduction

The decision tree is a powerful white-box classifier encompassing branches and nodes to deliver an interpretable model in the form of "if–else" rules. However, as most of the information is visible in the tree-like structure, the model is highly susceptible to privacy attacks. For example, the protocol type or port number attribute that branches out to or from the IP address node in the tree model may reveal the possibility of certain services being available on the particular machine [1].

Although Li and Sarkar [2] had proposed a privacy pruning model based on risk assessment, the approach is entirely different from the solution we proposed in this paper. Specifically, we utilise a case study in the network intrusion detection system (NIDS) domain to illustrate how we tackle the privacy issue as mentioned earlier. Furthermore, to the best of our knowledge, none of the previous works designed a privacy pruning solution from the perspective of IP truncation.

The IP address is viewed as personal data when its usage is combined with other information and can be used for profiling an individual [3,4]. For example, an attacker can deduce sensitive information such as individual online behaviours and personal communication from the IP address, and other information from the network traces [4]. IP truncation can be embedded into the C4.5 tree to anonymise any sensitive information [5,6].

However, this indicates that some information will be obscured, and careless handling of it will affect the performance of the C4.5 tree. In this work, we will examine the impacts of the IP truncation and C4.5 tree in the domain of NIDS.

2. Literature Review

A decision tree model often suffers from overfitting its training data due to its greedy approach in building the classifier. Pruning is often used to simplify an unpruned decision tree by removing the insignificance nodes, branches, or leaves [7–9]. Various pruning techniques have been proposed in the literature to yield a decision tree model with higher performance (i.e., classification accuracy) and better generalisation capability. Since the focus of this work is on tree pruning, some existing pruning strategies are discussed. The most common usage of pruning is to improve the decision tree model's classification accuracy (i.e., reduced classification error). In general, nodes or leaves will be pruned if the classification accuracy for the pruned version of the tree is better than the unpruned version. Reduced Error Pruning [10], Pessimistic Error Pruning [10], Error-based Pruning [11], Critical Value Pruning [12], Minimum Error Pruning [12], and Improved Expected Error Pruning [13] are some of the notable pruning algorithms that endeavour to minimise the error rate of the decision tree model. In the default settings, the C4.5 decision tree [11] uses Error-based Pruning (an improved version of Pessimistic Error Pruning).

A decision tree model can be viewed as a sequence of "if–else" rules flowing from the root node to the leaf node to provide a classification decision or regression result. Though maximising the classification accuracy is important, the structural complexity (i.e., number of nodes, branches, and leaves) of a tree should not be overlooked. This is because a tree with over-complicated tree structures would hinder the interpretation of the decision rules. Therefore, some researchers opt to build a simpler decision tree through the pruning mechanism to unleash the interpretability of a decision tree. On top of this, a simpler tree structure is more computational friendly. A few of the well-known pruning schemes that seek to reduce the complexity of the tree structure include Cost-Complexity Pruning [14], improved Cost-Complexity Pruning [15], and the Optimal Pruned Tree [16]. Among them, the Cost-Complexity Pruning was employed in the renowned Classification and Regression Trees (CART) developed by Breiman et al. [14].

Apart from pruning with a single objective such as maximising the classification accuracy or reducing the number of nodes, Osei-Bryson [17] pointed out another evaluation criteria—the stability and interpretability of a tree to assess the quality of the model. Fournier and Crémilleux [18] designed a Depth-Impurity Pruning based on the Impurity Quality Node index that considers the average distance of the data belonging to a leaf node to its centre of mass [19] and also the depth of a node. In contrast to Cost-Complexity Pruning [14], which only considers the number of nodes, Wei et al. [20] include the depth (i.e., level) of the tree and classification accuracy as the pruning evaluation criteria by utilising rough set theory [21].

In the purview of the unbalanced classes in most real-world classification problems, researchers had proposed a Cost-Sensitive Pruning technique to tackle this issue. It is very common to treat all classes as equally significant during the pruning process. However, this may not work perfectly for certain applications. In the medical field, a misclassification (i.e., false negative) in certain scenarios might be more expensive than a false positive. For instance, the consequence of identifying a patient with cancer as a healthy patient is more expensive than identifying a healthy patient as a patient diagnosed with cancer [22]. Thus, Knoll, Nakhaeizadeh, and Tausend [22] proposed a Cost-Sensitive Pruning that incorporates weights for each class in the dataset. In their work, they supplied the cost matrix directly to the algorithm. To evaluate the optimum ratio of misclassification costs (i.e., cost matrix), Bradley and Lovell [23] exploited the receiver operating characteristics curve, and Ting [24] incorporated instance weighting with a C4.5 decision tree to compute the cost matrix.

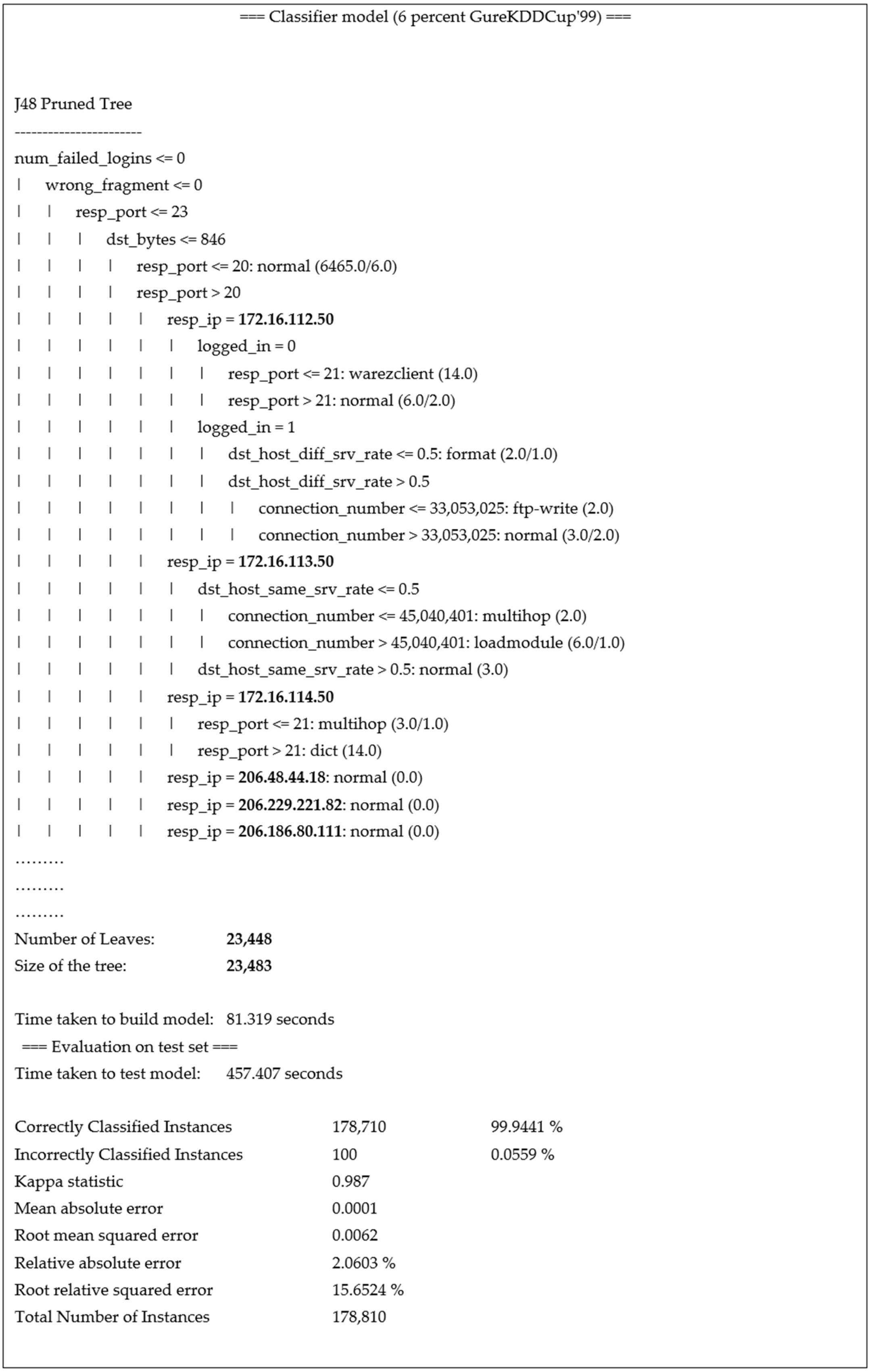
Decision tree models exploit the concepts of information gain to split the training dataset based on the features selected into a smaller group. In most cases, the leave nodes will only contain a very small subset of the training data. This scenario will lead to re-identification attacks through information linking [25], as described by Li and Sarkar [2]. Thus, Li and Sarkar [2] proposed a new pruning metric that evaluates each node's disclosure risks. They measured the risks by counting the number of data available in each node. A larger number of data would imply a lower risk, while a smaller number would suggest a higher risk of information disclosure. To prevent re-identification attacks, the authors introduced a data swapping procedure for the data found in each node. In this procedure, the majority class instances will be swapped with the minority class instances. To retain the classification performance, the superior probability after the data swapping process is retained by the majority class of each node.

From the literature review, we realise that only a minor group is working on privacy-oriented pruning, and there is room for improvement. Therefore, instead of weighting each node, we propose a new approach to prune the tree by truncating the value of the IP address in the application domain of NIDS. The methodology of this proposed work is presented in the next section.

3. Methodology

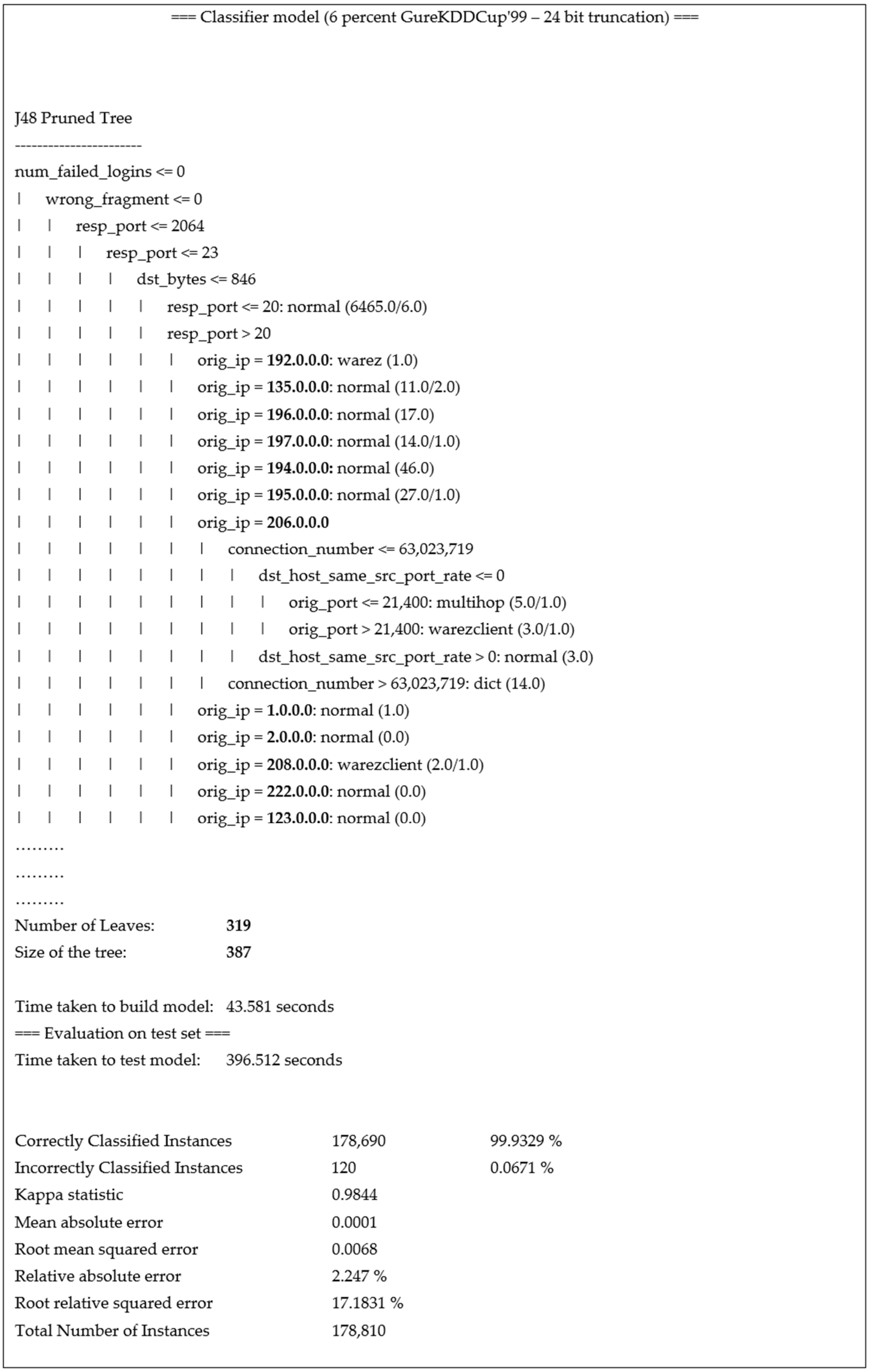
In this work, truncation [26] is performed on IP addresses in three ways: 8-bit, 16-bit, and 24-bit truncations. Given the original IP address as “192.168.123.121”, 8-bit truncation will zeroize the last byte (8-bit) of the IP address, thus representing it as “192.168.123.0”; 16-bit truncation will zeroize the last two bytes (16-bit) of the original IP address, representing it as “192.168.0.0”; whereas 24-bit truncation will zeroize the last three bytes (24-bit) of the original IP address, representing it as “192.0.0.0”. The goal of this proposed method is twofold: (1) to anonymise the real IP address and (2) to prune the original C4.5 decision tree. In this work, we have adopted the C4.5 decision tree from the Weka J48 decision tree package to run the experiment. It should be noted that the default J48 decision tree without any fine-tuning of parameters is utilised on all the experiments deliberated on in this paper.

To illustrate the impacts of decision tree pruning with IP truncation, part of the ASCII J48 (Weka implementation of C4.5) tree model built against the original 6-percent-GureKDDCup'99 dataset is shown in Figure 1. Similarly, the 24-bit truncation version of the same dataset trained with the identical dataset and model is presented in Figure 2. We can observe that all the IP addresses from the tree model in Figure 2 have been truncated by three bytes (24-bit), spawning a new set of IP addresses such as ”192.0.0.0”, “135.0.0.0”, “196.0.0.0”, etc. In both figures, the values in the bracket should be read as (total number of instances from the training distribution reaching the leaf, number of training instances that is incorrectly classified) [25]. For instance, “resp_port <= 20: normal (6465.0/6.0)” from Figure 2 denotes that 6465 number of training instances reached the leaf node, with 6 of them classified incorrectly [27]. To have a better insight into the proposed approach, we give the overall methodological diagram and pseudocode for the proposed IP truncation pruning algorithm in Figures 3 and 4. Note that 10-fold cross-validation is utilised in the absence of a testing dataset.



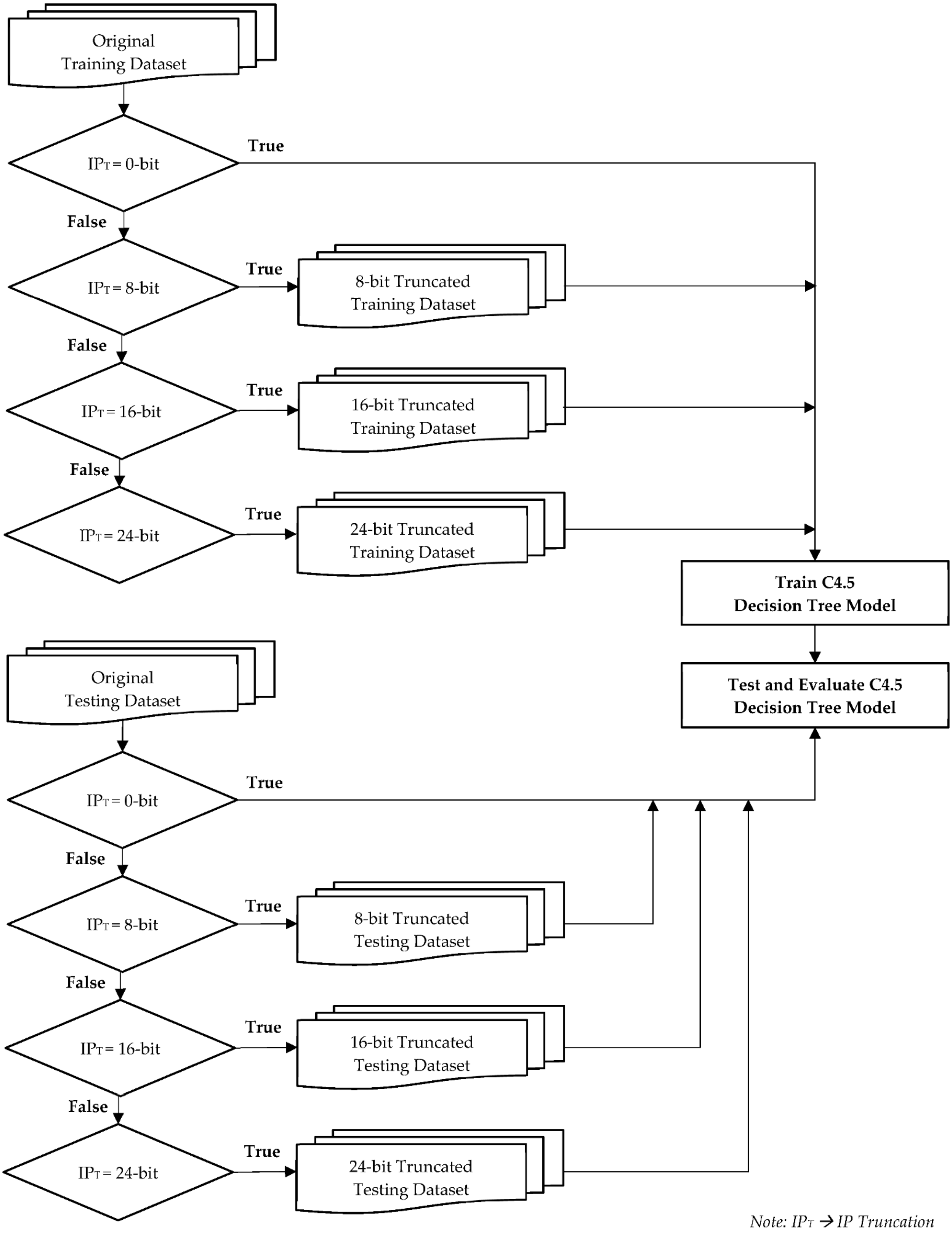
```
                              === Classifier model (6 percent GureKDDCup'99) ===


J48 Pruned Tree
-----------------------
num_failed_logins <= 0
|   wrong_fragment <= 0
|   |   resp_port <= 23
|   |   |   dst_bytes <= 846
|   |   |   |   resp_port <= 20: normal (6465.0/6.0)
|   |   |   |   resp_port > 20
|   |   |   |   |   resp_ip = 172.16.112.50
|   |   |   |   |   |   logged_in = 0
|   |   |   |   |   |   |   resp_port <= 21: warezclient (14.0)
|   |   |   |   |   |   |   resp_port > 21: normal (6.0/2.0)
|   |   |   |   |   |   logged_in = 1
|   |   |   |   |   |   |   dst_host_diff_srv_rate <= 0.5: format (2.0/1.0)
|   |   |   |   |   |   |   dst_host_diff_srv_rate > 0.5
|   |   |   |   |   |   |   |   connection_number <= 33,053,025: ftp-write (2.0)
|   |   |   |   |   |   |   |   connection_number > 33,053,025: normal (3.0/2.0)
|   |   |   |   |   resp_ip = 172.16.113.50
|   |   |   |   |   |   dst_host_same_srv_rate <= 0.5
|   |   |   |   |   |   |   connection_number <= 45,040,401: multihop (2.0)
|   |   |   |   |   |   |   connection_number > 45,040,401: loadmodule (6.0/1.0)
|   |   |   |   |   |   dst_host_same_srv_rate > 0.5: normal (3.0)
|   |   |   |   |   resp_ip = 172.16.114.50
|   |   |   |   |   |   resp_port <= 21: multihop (3.0/1.0)
|   |   |   |   |   |   resp_port > 21: dict (14.0)
|   |   |   |   |   resp_ip = 206.48.44.18: normal (0.0)
|   |   |   |   |   resp_ip = 206.229.221.82: normal (0.0)
|   |   |   |   |   resp_ip = 206.186.80.111: normal (0.0)
………
………
………
Number of Leaves:         23,448
Size of the tree:         23,483

Time taken to build model:  81.319 seconds
 === Evaluation on test set ===
Time taken to test model:   457.407 seconds

Correctly Classified Instances          178,710            99.9441 %
Incorrectly Classified Instances        100                0.0559 %
Kappa statistic                         0.987
Mean absolute error                     0.0001
Root mean squared error                 0.0062
Relative absolute error                 2.0603 %
Root relative squared error             15.6524 %
Total Number of Instances               178,810
```

Figure 1. ASCII J48 (Weka implementation of C4.5) Pruned Tree Model (partial), Performance Result on 6 Percent-GureKDDCup'99 (Original).



```
=== Classifier model (6 percent GureKDDCup'99 – 24 bit truncation) ===



J48 Pruned Tree
----------------------
num_failed_logins <= 0
|   wrong_fragment <= 0
|   |   resp_port <= 2064
|   |   |   resp_port <= 23
|   |   |   |   dst_bytes <= 846
|   |   |   |   |   resp_port <= 20: normal (6465.0/6.0)
|   |   |   |   |   resp_port > 20
|   |   |   |   |   |   orig_ip = 192.0.0.0: warez (1.0)
|   |   |   |   |   |   orig_ip = 135.0.0.0: normal (11.0/2.0)
|   |   |   |   |   |   orig_ip = 196.0.0.0: normal (17.0)
|   |   |   |   |   |   orig_ip = 197.0.0.0: normal (14.0/1.0)
|   |   |   |   |   |   orig_ip = 194.0.0.0: normal (46.0)
|   |   |   |   |   |   orig_ip = 195.0.0.0: normal (27.0/1.0)
|   |   |   |   |   |   orig_ip = 206.0.0.0
|   |   |   |   |   |   |   connection_number <= 63,023,719
|   |   |   |   |   |   |   |   dst_host_same_src_port_rate <= 0
|   |   |   |   |   |   |   |   |   orig_port <= 21,400: multihop (5.0/1.0)
|   |   |   |   |   |   |   |   |   orig_port > 21,400: warezclient (3.0/1.0)
|   |   |   |   |   |   |   |   dst_host_same_src_port_rate > 0: normal (3.0)
|   |   |   |   |   |   |   connection_number > 63,023,719: dict (14.0)
|   |   |   |   |   |   orig_ip = 1.0.0.0: normal (1.0)
|   |   |   |   |   |   orig_ip = 2.0.0.0: normal (0.0)
|   |   |   |   |   |   orig_ip = 208.0.0.0: warezclient (2.0/1.0)
|   |   |   |   |   |   orig_ip = 222.0.0.0: normal (0.0)
|   |   |   |   |   |   orig_ip = 123.0.0.0: normal (0.0)
………
………
………
Number of Leaves:            319
Size of the tree:            387

Time taken to build model:   43.581 seconds
=== Evaluation on test set ===
Time taken to test model:    396.512 seconds



Correctly Classified Instances          178,690          99.9329 %
Incorrectly Classified Instances        120              0.0671 %
Kappa statistic                         0.9844
Mean absolute error                     0.0001
Root mean squared error                 0.0068
Relative absolute error                 2.247 %
Root relative squared error             17.1831 %
Total Number of Instances               178,810
```

Figure 2. ASCII J48 (Weka implementation of C4.5) Pruned Tree Model (partial), Performance Result on 6 Percent-GureKDDCup’99 (24-bit Truncation).


Original Training Dataset
IPT = 0-bit
True
False
IPT = 8-bit
True
8-bit Truncated Training Dataset
False
IPT = 16-bit
True
16-bit Truncated Training Dataset
False
IPT = 24-bit
True
24-bit Truncated Training Dataset
Train C4.5 Decision Tree Model
Test and Evaluate C4.5 Decision Tree Model
Original Testing Dataset
IPT = 0-bit
True
False
IPT = 8-bit
True
8-bit Truncated Testing Dataset
False
IPT = 16-bit
True
16-bit Truncated Testing Dataset
False
IPT = 24-bit
True
24-bit Truncated Testing Dataset

Note: IPT → IP Truncation

Figure 3. Methodology Diagram of the Proposed IP Truncation Pruning Algorithm.

```
Pseudocode of IP Truncation Pruning Algorithm
Input:        $D_{Train}$,   Original Training Dataset;
              $D_{Test}$,    Original Testing Dataset;
              $IP_T$,        Truncation Bit
Output:       C4.5 Decision Tree Pruned with IP Truncation
Procedure:
    if $IP_T$ equal to 0-bit:          retain the original $D_{Train}$ and $D_{Test}$
    elseif $IP_T$ equal to 8-bit:      perform 8-bit Truncation on $D_{Train}$ and $D_{Test}$
    elseif $IP_T$ equal to 16-bit:     perform 16-bit Truncation on $D_{Train}$ and $D_{Test}$
    elseif $IP_T$ equal to 24-bit:     perform 24-bit Truncation on $D_{Train}$ and $D_{Test}$
    Train the C4.5 Decision Tree with $D_{Train}$
    Test the C4.5 Decision Tree with $D_{Test}$
    Evaluate the performances of the Decision Trees with several evaluation metrics
```

Figure 4. Pseudocode of the Proposed IP Truncation Pruning Algorithm.

4. Experimental Empirical Studies

The examination is tested on a 6-percent-GureKDDCup'99 [28,29] NIDS dataset. This dataset is suitable because it contains IP addresses, which the IP truncation method will be applied to. To avoid the biasness or cherry-picking of the features, this work intentionally opted out of feature extraction, and no feature enhancement was performed on the dataset.

Hence, all the 47 attributes including connection_number, start_time, orig_port, resp_port, orig_ip, resp_ip, duration, protocol_type, service, flag, src_bytes, dst_bytes, land, wrong_fragment, urgent, hot, num_failed_logins, logged_in, num_compromised, root_shell, su_attempted, num_root, num_file_creations, num_shells, num_access_files, num_outbound_cmds, is_hot_login, is_guest_login, count, srv_count, serror_rate, srv_serror_rate, rerror_rate, srv_rerror_rate, same_srv_rate, diff_srv_rate, srv_diff_host_rate, dst_host_count, dst_host_srv_count, dst_host_same_srv_rate, dst_host_diff_srv_rate, dst_host_same_src_port_rate, dst_host_srv_diff_host_rate, dst_host_serror_rate, dst_host_srv_serror_rate, dst_host_rerror_rate, dst_host_srv_error_rate are utilised in the experimental procedure. Details of the attributes can be referred from the GureKDDCup'99 documentation [28,29].

As the dataset has not been segregated into a training and testing distribution by the authors, the classification performance of the proposed method will be evaluated based on 10-fold cross-validation to provide a fair comparison between the models.

4.1. Experimental Results

To gauge the usability of a decision tree model, the decision tree structure (i.e., number of nodes) and the classification accuracy attained before and after the application of IP truncations are compared. The number of nodes is selected as one of the evaluation criteria since the interpretability of a decision tree greatly depends on the tree structure. For instance, a very complex decision tree structure having a vast number of nodes is harder to be understood when compared to a simpler tree with a smaller number of nodes.

We present the empirical results for each model (i.e., original, 8-bit truncation, 16-bit truncation, and 24-bit truncation) in Table 1 with crucial evaluation metrics such as classification accuracy, true positive rate, false positive rate, true negative rate, false negative rate, the number of nodes found in the tree models, and the number of leaves present in the tree models.

Table 1. Performance Comparison of Each Model in 6 Percent-GureKDDCup'99.

		IP Truncation		
Criteria	**Original**	**8-Bit**	**16-Bit**	**24-Bit**
Classification Accuracy (%)	**99.9441**	99.9323	99.9279	99.9329
Accuracy Degradation	-	**−0.0118**	**−0.0162**	**−0.0112**
True Positive Rate	**0.9994**	0.9993	0.9993	0.9993
False Positive Rate	**0.0137**	0.0224	0.0216	0.0189
True Negative Rate	**0.9863**	0.9776	0.9784	0.9811
False Negative Rate	**0.0006**	0.0007	0.0007	0.0007
Number of Nodes	23,483	20,663	2156	**387**
Reduction in Number of Nodes	-	**2820**	**21,327**	**23,096**
Reduction in Nodes (%)	-	12.0087	90.8189	**98.3520**
Number of Leaves	23,448	20,621	2094	319
Reduction in Number of Leaves	-	**2827**	**21,354**	**23,129**
Reduction in Leaves (%)	-	12.0565	91.0696	**98.6395**
Number of Source IP Address	99	72	64	27
Number of Destination IP address	5801	4057	1730	72
Total Number of IP address	5900	4129	1794	99
Reduction in Number of IP Address	-	**1771**	**4106**	**5801**
Reduction in Number of IP Address (%)	-	30.0169	69.5932	**98.3220**

4.2. Discussions and Findings

Affected by the IP truncation concealing procedure, it is speculated that this process might be reducing the accuracy of the original decision tree due to information loss. However, the experimental results prove otherwise since the difference in accuracy between the original and truncated versions are remarkably insignificant (on average ~0.01% of degradation), as tabulated in Table 1. Similarly, although the original model performance in terms of true positive rate, false positive rate, true negative rate, and false negative rate is superior when compared to the other models, the difference is miniature.

Interestingly, the total number of nodes in the decision tree structure is observed to be considerably decreased after applying IP truncation (i.e., 23,096 or 98.3520% of nodes are pruned from the original tree when 24-bit truncation is adopted) based on the results shown in Table 1. This scenario signifies that the truncation scheme can simplify the tree structure and provide additional privacy protection without deteriorating the classification performance of the decision tree at the same time.

Akin to the observation in the number of nodes, we also noticed a significant reduction in the number of leaves. In particular, the number of leaves is reduced by 21,354 (91.0696%) when 16-bit truncation is adopted and 23,129 (98.6395%) when 24-bit truncation is applied.

To better understand the differences between the truncation methods, we have also presented the total number of distinct source IP addresses and destination IP addresses in Table 1. The highest number of nodes and leaves are pruned when 24-bit truncation is applied. This can be substantiated by the total reduction in IP address from 5900 to 99. A total of 5801 or 98.3220% of IP addresses have been reduced as compared to the original model when 24-bit truncation is employed.

Based on the empirical results presented in this section, it is safe to assume that the truncation methods are suitable to be adopted as a pruning strategy to reduce the complexity of the tree structure while maintaining an adequate classification performance and preserving privacy simultaneously.

5. Extended Experimental Empirical Studies

To further corroborate the claims made in Section 4, extended empirical studies performed on the full version of three NIDS datasets are substantiated in this section. The three NIDS datasets include (i) GureKDDCup'99 [28,29], (ii) UNSW-NB15 [30], and CIDDS-

001 [31]. As the full version of these three datasets was supplied in a weekly fashion by the authors, we adopted a distinct approach to split the training and testing data following each of the weeks, as depicted in Table 2. Due to the over-optimistic results produced by cross-validation shown in the previous section, this approach was advocated by Ring et al. [32] and Al Tobi and Duncan [33]. Akin to the 6-percent-GureKDDCup’99, the three NIDS datasets are chosen as they contained the mandatory IP truncation features: the source IP address and destination IP address. More details of the datasets, data cleansing, and data preparation can be found in our previous work [34].

Table 2. Distribution of Training and Testing Sets for the full version of GureKDDCup’99, UNSW-NB15, and CIDDS-001.

Dataset	Training Set Distribution	Testing Set Distribution
GureKDDCup’99 (full)	Week 1 Week 1~2 Week 1~3 Week 1~4 Week 1~5 Week 1~6	Week 2~7 Week 3~7 Week 4~7 Week 5~7 Week 6~7 Week 7
UNSW-NB15 (full)	Week 1 Week 1~2 Week 1~3	Week 2~4 Week 3~4 Week 4
CIDDS-001	Week 1	Week 2

5.1. Extended Empirical Experimental Results

For the sake of simplicity, only the classification accuracy (i.e., model performance) and the number of nodes (i.e., tree structure) will be discussed and analysed in the subsequent section. Table 3 tabulates the accuracy performance of the decision tree model trained with different train–test distributions incorporated with or without the IP truncation. The number of nodes present in the model after the pruning process is delivered in Table 4. From the tables, we can notice that the obtained empirical results are very encouraging as most tree models pruned with IP truncation can perform almost on a par with the original model.

Table 3. Classification Accuracy (%) for each C4.5 Tree Model in Full GureKDDCup’99, UNSW-NB15, and CIDDS-001.

Dataset		IP Truncation			
Train	**Test**	**Original**	**8-Bit**	**16-Bit**	**24-Bit**
GureKDDCup					
1	2~7	38.0293	38.0293	38.0293	38.0293
1~2	3~7	33.1808	33.1808	33.1808	33.1808
1~3	4~7	**82.8830**	82.8707	82.8741	82.8794
1~4	5~7	88.1692	88.4873	89.0633	**89.0881**
1~5	6~7	88.8412	88.8432	88.8412	**88.8490**
1~6	7	**99.9526**	99.9508	99.8981	99.7670
UNSW-NB15					
1	2~4	96.5079	96.8354	**96.8500**	**96.8500**
1~2	3~4	96.6146	**96.6783**	**96.6783**	**96.6783**
1~3	4	96.7751	**96.9755**	**96.9755**	**96.9755**
CIDDS-001					
1	2	**94.4539**	93.4896	92.4927	92.4927

Table 4. Number of Nodes for each C4.5 Tree Model in Full GureKDDCup'99, UNSW-NB15, and CIDDS-001.

Dataset		IP Truncation			
Train	**Test**	**Original**	**8-Bit**	**16-Bit**	**24-Bit**
GureKDDCup					
1	2~7	5	5	5	5
1~2	3~7	2990	2047	46	**41**
1~3	4~7	19,615	3522	1807	**367**
1~4	5~7	102,422	37,399	4205	**587**
1~5	6~7	175,904	53,695	4905	**867**
1~6	7	318,262	44,350	7313	**1096**
UNSW-NB15					
1	2~4	3744	**3126**	**3116**	**3116**
1~2	3~4	8762	**5443**	**5426**	**5426**
1~3	4	18,787	**9639**	**9625**	**9625**
CIDDS-001					
1	2	8766	720	**612**	**612**

5.2. Discussion and Analysis on the Extended Empirical Experimental Results

Similar to the observation presented in Section 4, the classification performance difference between the truncated models and original model is not significant, with, at most, 2% of degradation across the three datasets based on the classification accuracy difference shown in Table 5. However, it is interesting to witness that some tree models utilising the IP truncation pruning mechanism could perform better than the original model. For example, when the model is trained with GureKDDCup'99 (Week 1~4) and tested with GureKDDCup'99 (Week 5~7), the proposed model integrated with the 24-bit truncation gains 0.9189% accuracy compared with the original model. This scenario assumes that the nodes pruned by truncation can produce a tree with better generalisation capability.

Table 5. Classification Accuracy (%) Difference for each C4.5 Model in Full GureKDDCup'99, UNSW-NB15, and CIDDS-001.

Dataset		IP Truncation		
Train	**Test**	**8-Bit**	**16-Bit**	**24-Bit**
GureKDDCup				
1	2~7	0.0000	0.0000	0.0000
1~2	3~7	0.0000	0.0000	0.0000
1~3	4~7	−0.0123	−0.0088	−0.0036
1~4	5~7	**0.3181**	**0.8942**	**0.9189**
1~5	6~7	**0.0020**	0.0000	**0.0079**
1~6	7	−0.0018	−0.0545	−0.1856
UNSW-NB15				
1	2~4	**0.3275**	**0.3421**	**0.3421**
1~2	3~4	**0.0637**	**0.0637**	**0.0637**
1~3	4	**0.2004**	**0.2004**	**0.2004**
CIDDS-001				
1	2	−0.9642	−1.9611	−1.9611

ACC: Classification Accuracy (%) = Accuracy (IP Truncated) − Accuracy (Original Data); (+) indicates Improvement in Accuracy; (−) indicates Reduction in Accuracy.

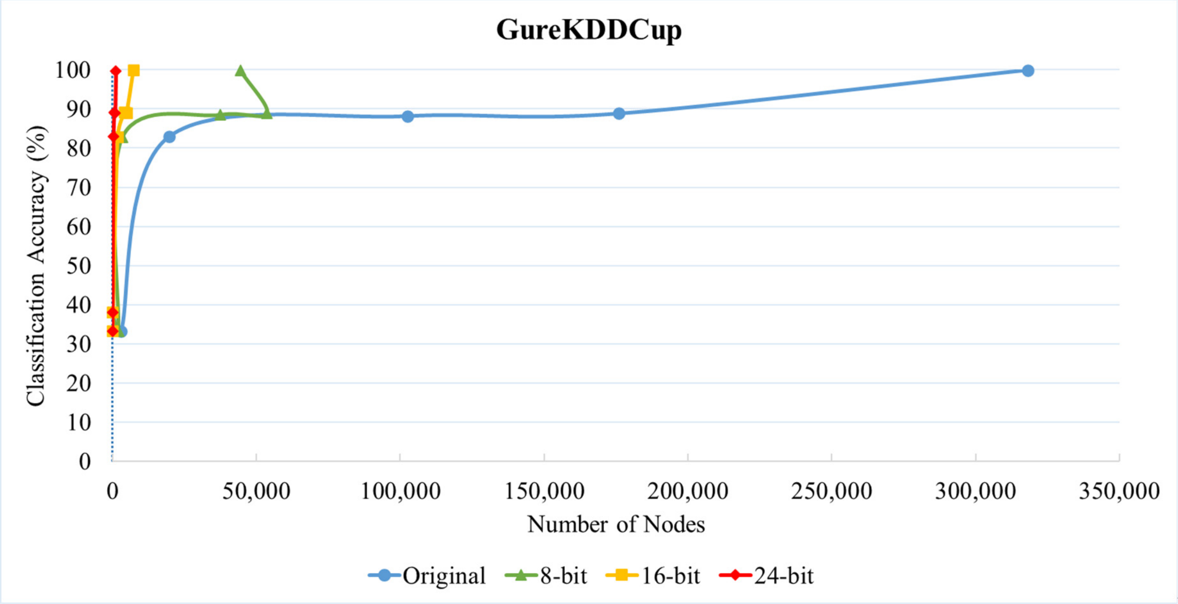
For the tree's structure, the number of nodes realised by each model is tabulated in Table 6. From the result, we can observe that the number of nodes is significantly reduced in all three datasets when 8-bit, 16-bit, and 24-bit truncations are adopted. These results further justify that the IP truncation is an effective strategy to prune the tree to create a simpler tree structure.

Table 6. Performance Comparison (Number of Nodes) for each C4.5 Model in Full GureKDDCup'99, UNSW-NB15, and CIDDS-001.

Dataset		IP Truncation					
		8-Bit		16-Bit		24-Bit	
Train	Test	No. Nodes [1]	Reduction (%)	No. Nodes [1]	Reduction (%)	No. Nodes [1]	Reduction (%)
GureKDDCup							
1	2~7	0	0.00	0	0.00	0	0.00
1~2	3~7	−943	−31.54	−2944	−98.46	**−2949**	**−98.63**
1~3	4~7	−16,093	−82.04	−17,808	−90.79	**−19,248**	**−98.13**
1~4	5~7	−65,023	−63.49	−98,217	−95.89	**−101,835**	**−99.43**
1~5	6~7	−122,209	−69.47	−170,999	−97.21	**−175,037**	**−99.51**
1~6	7	−273,912	−86.06	−310,949	−97.70	**−317,166**	**−99.66**
UNSW-NB15							
1	2~4	−618	−16.51	**−628**	**−16.77**	**−628**	**−16.77**
1~2	3~4	−3319	−37.88	**−3336**	**−38.07**	**−3336**	**−38.07**
1~3	4	−9148	−48.69	**−9162**	**−48.77**	**−9162**	**−48.77**
CIDDS-001							
1	2	−8046	−91.79	**−8154**	**−93.02**	**−8154**	**−93.02**

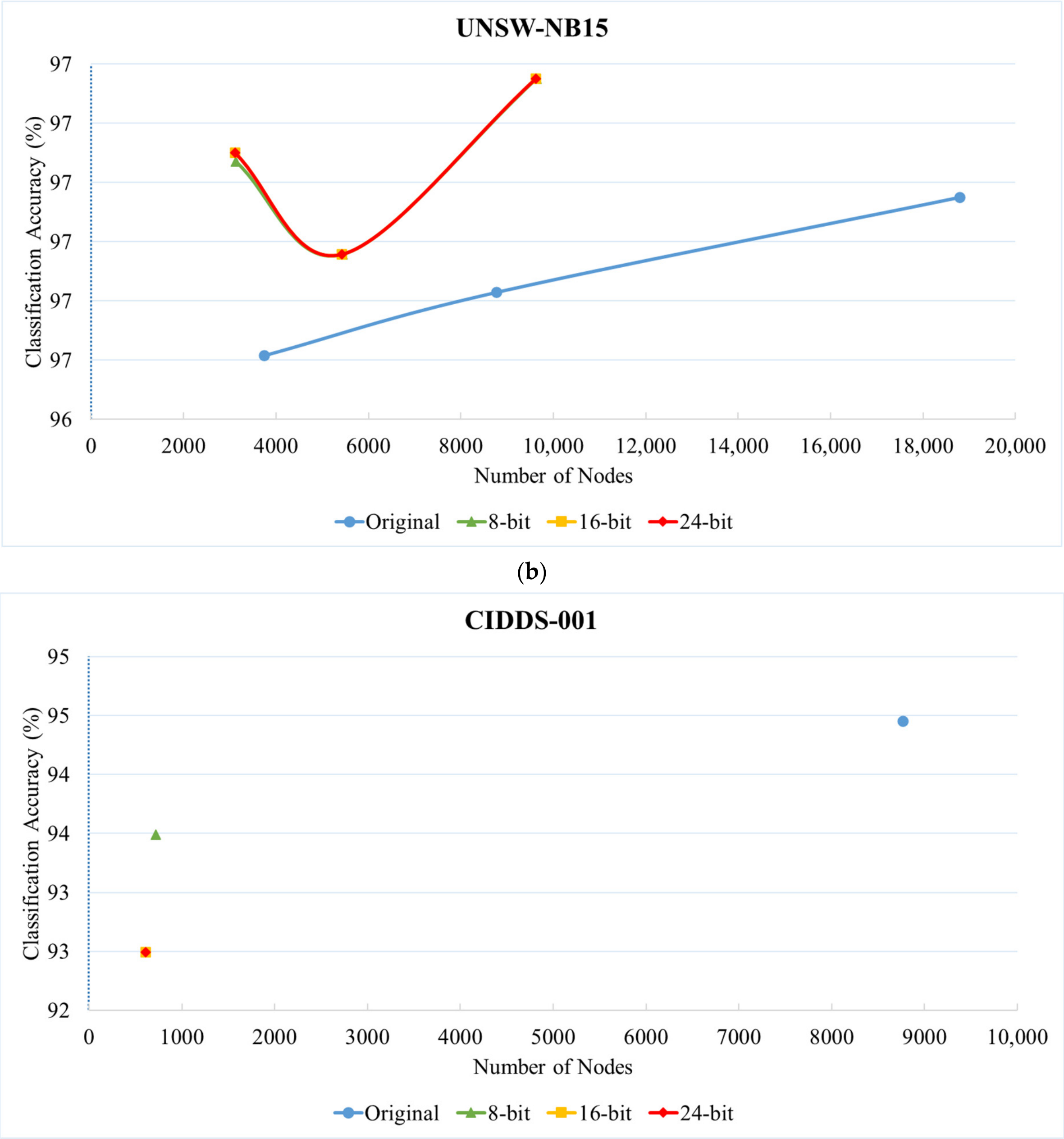
[1] No. Nodes: Number of Nodes = No. Nodes (IP Truncated) − No. Nodes (Original Data); (+) denotes a more Complex Tree; (−) denotes a Simpler Tree.

By plotting the classification accuracy against the number of nodes in Figure 5, it can be evidently seen that the truncation approach can prune the nodes efficiently without sacrificing a good classification accuracy. Interestingly, the performance of the truncated models containing lesser nodes in GureKDDCup'99 and UNSB-SW15 are either on a par with or superior when compared to the original models. Although the model's classification accuracy slightly deteriorated by ~2% in CIDDS-001, it is compensated with a simpler tree containing only ~700 nodes instead of 8766 nodes from the original model.



(**a**)

Figure 5. *Cont.*



(**b**)

(**c**)

Figure 5. Classification Accuracy (%) (*Y*-axis) Versus the Number of Nodes (*X*-axis) in (**a**) GureKDD-Cup'99, (**b**) UNSW-NB15, (**c**) CIDDS-001.

From Table 5, we noticed that the classification performance of the tree models is identical when 16-bit or 24-bit truncation is applied in UNSW-NB15 and CIDDS-001. At the same time, we also observed that the number of nodes remains constant in accordance with the results in Table 6. To further understand this finding, the total number of unique IP addresses, including the source and destination, is calculated and organised in Table 7. Therefore, we can identify that the total number of IP addresses remains unchanged in UNSW-NB15 and CIDDS-001 when 16-bit or 24-bit truncation is applied. This may be the key reason leading to the identical performance achieved by the models.

Table 7. Total Number of Distinct IP addresses (Sum of both Source and Destination IP) for each Train–Test Distribution in Full GureKDDCup'99, UNSW-NB15, and CIDDS-001.

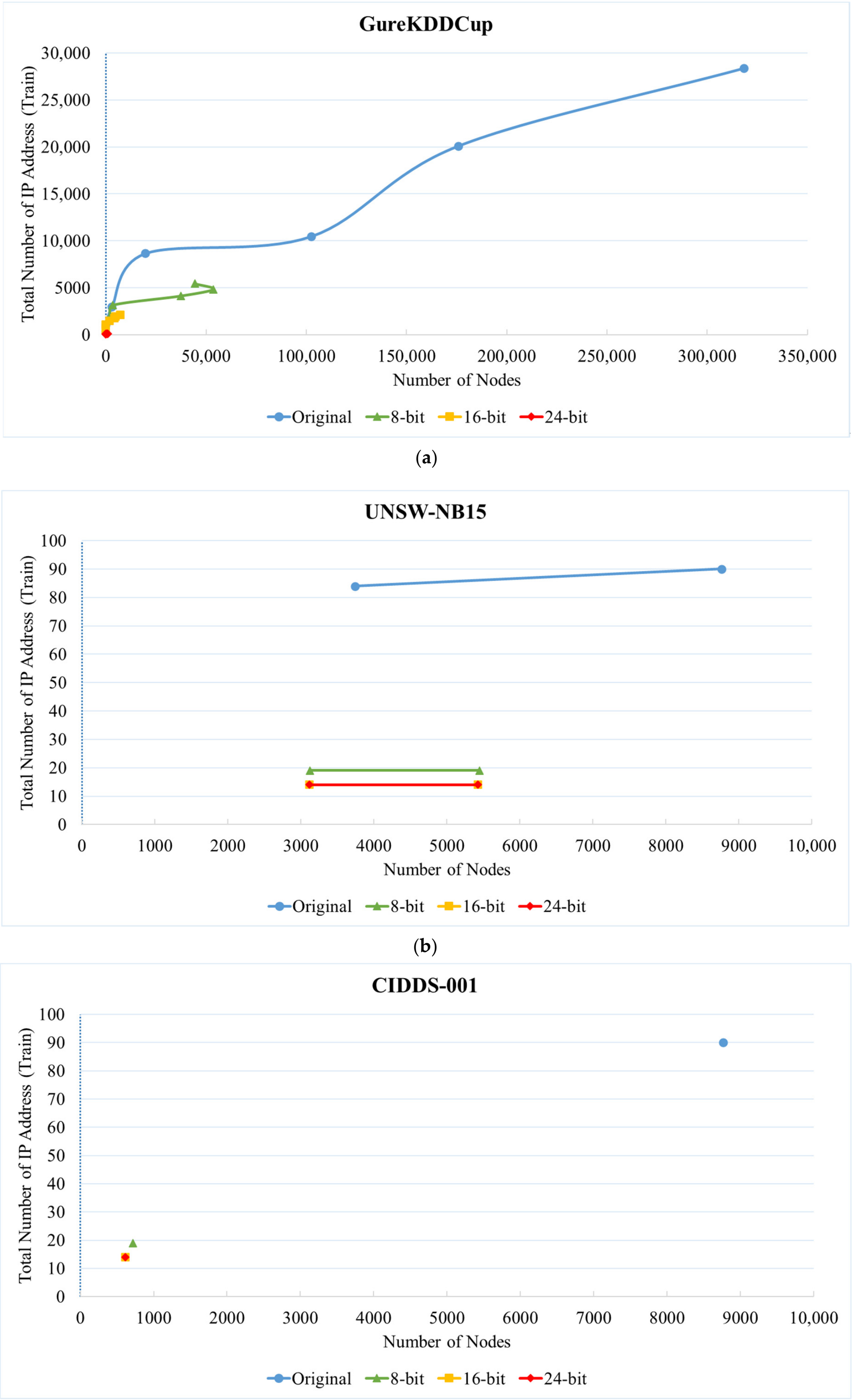
Dataset		IP Truncation							
Train	Test	Original		8-Bit		16-Bit		24-Bit	
		Train	Test	Train	Test	Train	Test	Train	Test
GureKDDCup									
1	2~7	1083	29,052	759	5828	522	2158	**68**	**104**
1~2	3~7	2960	27,832	2017	5138	1103	2013	**78**	**104**
1~3	4~7	8661	23,793	3156	4356	1497	1841	**90**	**102**
1~4	5~7	10,472	22,481	4133	3505	1783	1624	**98**	**100**
1~5	6~7	20,106	20,763	4855	2610	1952	1334	**102**	**96**
1~6	7	28,366	4001	5442	1617	2093	932	**104**	**84**
UNSW-NB15									
1	2~4	84	86	19	16	**14**	**11**	**14**	**11**
1~2	3~4	90	86	19	16	**14**	**11**	**14**	**11**
1~3	4	90	81	19	16	**14**	**11**	**14**	**11**
CIDDS-001									
1	2	827	76	16	18	**10**	**12**	**10**	**12**

In Table 8, we present the total number of reductions in IP addresses when different truncation approaches are applied in the three datasets. Since the tree model only utilised the number of unique IP addresses from the training distribution as the training attributes to build the tree, we deduced that the total number of unique IP addresses found in the training distribution will have a particularly significant relationship with the number of nodes. Referring to Table 8, we noticed a huge amount of IP address reduction in GureKDDCup'99 when 24-bit truncation is applied on the Week 1~6 training distribution, whereby a total of 28,262 (99.63%) unique IP addresses are removed. By eliminating a tremendous amount of unique IP addresses, the number of nodes is also significantly pruned by 317,166 nodes (refer Table 6) compared to the original model. Figure 6 is plotted to illustrate the relationship between the number of nodes and the total number of unique IP addresses from the training distribution. We can clearly perceive that as the number of unique IP addresses increases, the number of nodes produced by the tree model will also increase. This undoubtedly justifies the noticeable effects of IP truncation when applied on GureKDDCup'99 since it contains the largest number of unique IP addresses.

Table 8. Performance Comparison (Total Number of Distinct IP address) for each C4.5 Model in Full GureKDDCup'99, UNSW-NB15, and CIDDS-001.

Dataset		IP Truncation											
Train	Test	8-Bit				16-Bit				24-Bit			
		Train		Test		Train		Test		Train		Test	
		R. IP	(%)	R. IP	(%)	R. IP	(%)	R. IP	(%)	R. IP	(%)	R. IP	(%)
GureKDDCup													
1	2~7	−324	−29.92	−23,224	−79.94	−561	−51.80	−26,894	−92.57	**−1015**	**−93.72**	**−28,948**	**−99.64**
1~2	3~7	−943	−31.86	−22,694	−81.54	−1857	−62.74	−25,819	−92.77	**−2882**	**−97.36**	**−27,728**	**−99.63**
1~3	4~7	−5505	−63.56	−19,437	−81.69	−7164	−82.72	−21,952	−92.26	**−8571**	**−98.96**	**−23,691**	**−99.57**
1~4	5~7	−6339	−60.53	−18,976	−84.41	−8689	−82.97	−20,857	−92.78	**−10,374**	**−99.06**	**−22,381**	**−99.56**
1~5	6~7	−15,251	−75.85	−18,153	−87.43	−18,154	−90.29	−19,429	−93.58	**−20,004**	**−99.49**	**−20,667**	**−99.54**
1~6	7	−22,924	−80.82	−2384	−59.59	−26,273	−92.62	−3069	−76.71	**−28,262**	**−99.63**	**−3917**	**−97.90**
UNSW-NB15													
1	2~4	−65	−77.38	−70	−81.40	**−70**	**−83.33**	**−75**	**−87.21**	**−70**	**−83.33**	**−75**	**−87.21**
1~2	3~4	−71	−78.89	−70	−81.40	**−76**	**−84.44**	**−75**	**−87.21**	**−76**	**−84.44**	**−75**	**−87.21**
1~3	4	−71	−78.89	−65	−80.25	**−76**	**−84.44**	**−70**	**−86.42**	**−76**	**−84.44**	**−70**	**−86.42**
CIDDS-001													
1	2	−811	−98.07	−58	−76.32	**−817**	**−98.79**	**−64**	**−84.21**	**−817**	**−98.79**	**−64**	**−84.21**

R. IP: Reduction in IP = Total Number of IP (IP Truncated) − Total Number of IP (Original Data).



(**a**)

(**b**)

(**c**)

Figure 6. Total Number of IP Addresses in the Training Set (*Y*-axis) Versus the Number of Nodes (*X*-axis) in (**a**) GureKDDCup'99, (**b**) UNSW-NB15, (**c**) CIDDS-001

The empirical results substantiated in this section further uphold the claims made in Section 4. This substantially demonstrates that the adoption of the truncation approach to pruning the tree model will not cause a substantial decline in the classification performance of the model. In short, the truncation strategy is a suitable method to preserve privacy and reduce the complexity of the tree concurrently without sacrificing the classification performance.

6. Performance Comparison Using Wilcoxon Signed-Rank Test

To further evaluate and ensure that the adoption of the proposed pruning does not deteriorate the performance of the original classifier, we have additionally performed a Wilcoxon Signed-Rank test. A two-tailed test with a significance level of 0.05 is conducted based on the original classification accuracy against the truncated classification accuracy (8-bit, 16-bit, and 24-bit) obtained from Tables 1 and 3.

According to Table 9, all the tests show that the results are insignificant. In other words, the finding affirms that the performance achieved by the proposed approach is identical to the original classifier. However, we also notice that the classification accuracy attained by the proposed method in some scenarios surpasses the original classifier. Specifically, the accuracy is improved when the full GureKDDCup'99 is trained (Week 1~4; Week 1~5) and tested against (Week 5~7; Week 6~7) for all 8-bit, 16-bit, and 24-bit truncation. Additionally, all the models that exploited the IP truncated pruning approaches outperformed the original classifier in the UNSW-NB15 datasets. Details of the performance comparison can be found in Table 5.

Table 9. Performance Comparison of Pruning with IP Truncation Versus Original Decision Tree using Wilcoxon Signed-Rank Test.

Treatment 1	Treatment 2	Results	z-Value	w-Value
Original Decision Tree	8-bit Truncation	**Not Significant**	−0.6516	17
	16-bit Truncation	**Not Significant**	−0.5601	14
	24-bit Truncation	**Not Significant**	−0.05331	18

Significance Level: 0.05; Hypothesis: Two-Tailed.

These test results further confirm the claims made in Sections 4.2 and 5.2. We can comprehend that the adoption of IP truncation can simplify the tree structure through pruning and preserve privacy at the same time without deteriorating the performance of the tree classifier.

7. Conclusions and Research Limitation

The proposed pruning method, which is based on IP truncation, spells two main advantages: (1) the IP address can be anonymised to prevent any potential user profiling, and (2) the number of nodes in a C4.5 tree is tremendously reduced to make the rule interpretation possible while maintaining the classification accuracy.

A slight accuracy degradation is observed from the results presented after we applied the truncation in most scenarios. However, it is safe to assume that the truncation can prune the decision tree effectively without much affecting the performance of the tree classifier, and merely a very small performance degradation is observed (refer to Tables 1 and 5) after the truncation. Based on the empirical results presented in Table 5, it is also interesting to observe that the classification accuracy of the truncated model outperforms the original model in some situations. It can be postulated that the IP truncation can group similar features for building a more generalised tree model.

However, masking the IP address for the sake of privacy will also hinder its interpretability when we want to investigate an incident for an exact node (i.e., IP address or specific device). This can be viewed as a trade-off of privacy. Based on the empirical results attained, we believe that the proposed IP truncation-based pruning can preserve the utility and privacy simultaneously.

8. Future Works

In future, other network anonymisation techniques and machine learning classifiers will be explored and investigated to further assess the relationship between them. On the other hand, the similar analysis procedure demonstrated in this paper can also be adopted in other domains to examine the pruning effect of a decision tree resulting from the application of data privacy solutions. Nevertheless, future experiments in the Botnet [35] application may also be considered to exclude a list of Botnet IP addresses from the IP truncation anonymisation algorithm since they can be regarded as public information.

Author Contributions: Y.J.C.: conceptualisation, data curation, formal analysis, investigation, methodology, writing—original draft. S.Y.O.: funding acquisition, project administration, supervision, writing—review and editing. K.-S.W. and Y.H.P.: supervision, writing—review and editing. N.L.: data curation, validation. All authors have read and agreed to the published version of the manuscript.

Funding: This research work was supported by a Fundamental Research Grant Scheme (FRGS) under the Ministry of Education and Multimedia University, Malaysia (Project ID: MMUE/160029), and Korea Foundation of Advanced Studies (ISEF).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: GureKDDCup'99, Perona et al. [28,29]. Publicly available from: http://aldapa.eus/res/gureKddcup/ (15 December 2021); UNSW-NB15, Moustafa and Slay [30]. Publicly available from: https://doi.org/10.26190/5d7ac5b1e8485 (15 December 2021); CIDDS-001, Ring et al. [31]. Publicly available from: https://www.hs-coburg.de/forschung/forschungsprojekte-oeffentlich/informationstechnologie/cidds-coburg-intrusion-detection-data-sets.html (15 December 2021).

Conflicts of Interest: No competing interest was disclosed.

References

1. Chew, Y.J.; Ooi, S.Y.; Wong, K.-S.; Pang, Y.H. Decision Tree with Sensitive Pruning in Network-based Intrusion Detection System. In *Computational Science and Technology*; Alfred, R., Lim, Y., Haviluddin, H., On, C.K., Eds.; Springer: Berlin/Heidelberg, Germany, 2020; Volume 603, pp. 1–10. [CrossRef]
2. Li, X.-B.; Sarkar, S. Against Classification Attacks: A Decision Tree Pruning Approach to Privacy Protection in Data Mining. *Oper. Res.* **2009**, *57*, 1496–1509. [CrossRef]
3. Yurcik, W. Safely Sharing Data Between CSIRTs: The SCRUB * Security Anonymization Tool Infrastructure the SCRUB * Architecture. Available online: https://www.first.org/resources/papers/conference2008/yurcik-William-slides.pdf (accessed on 16 December 2021).
4. Riboni, D.; Villani, A.; Vitali, D.; Bettini, C.; Mancini, L.V. Obfuscation of sensitive data for increamental release of Network Flows. *IEEE/ACM Trans. Netw.* **2015**, *23*, 2372–2380. [CrossRef]
5. Slagell, A.; Lakkaraju, K.; Luo, K. FLAIM: A Multi-level Anonymization Framework for Computer and Network Logs. In Proceedings of the LISA '06: 20th Large Installation System Administration Conference, Washington, DC, USA, 3–8 December 2006; USENIX Association: Berkeley, CA, USA, 2006.
6. Yurcik, W.; Woolam, C.; Hellings, G.; Khan, L.; Thuraisingham, B. SCRUB-tcpdump: A multi-level packet anonymizer demonstrating privacy/analysis tradeoffs. In Proceedings of the 3rd International Conference on Security and Privacy in Communications Networks and the Workshops-SecureComm 2007, Nice, France, 17–21 September 2007; pp. 49–56. [CrossRef]
7. Mingers, J. An empirical comparison of pruning methods for decision tree induction. *Mach. Learn.* **1989**, *4*, 227–243. [CrossRef]
8. Patil, D.D.; Wadhai, V.M.; Gokhale, J.A. Evaluation of Decision Tree Pruning Algorithms for Complexity and Classification Accuracy. *Int. J. Comput. Appl.* **2010**, *11*, 975–8887. [CrossRef]
9. Patel, N.; Upadhyay, S. Study of Various Decision Tree Pruning Methods with their Empirical Comparison in WEKA. *Int. J. Comput. Appl.* **2012**, *60*, 20–25. [CrossRef]
10. Quinlan, J.R. Simplifying decision trees. *Int. J. Man. Mach. Stud.* **1987**, *27*, 221–234. [CrossRef]
11. Quinlan, J.R. *C4.5: Programs for Machine Learning*; Morgan Kaufmann Publishers Inc.: San Mateo, CA, USA, 1993; ISBN 1-55860-238-0.
12. Mingers, J. An empirical comparison of selection measures for decision tree induction. *Mach. Learn.* **1989**, *3*, 319–342. [CrossRef]
13. Zhang, Y.; Chi, Z.X.; Wang, D.G. Decision tree's pruning algorithm based on deficient data sets. In Proceedings of the Sixth International Conference on Parallel and Distributed Computing Applications and Technologies (PDCAT'05), Dalian, China, 5–8 December 2005; pp. 1030–1032. [CrossRef]
14. Breiman, L.; Friedman, J.; Stone, C.J.; Olshen, R.A. *Classification and Regression Trees*; Taylor & Francis: Boca Raton, FL, USA, 1984.

15. Gelfand, S.B.; Ravishankar, C.S.; Delp, E.J. An Iterative Growing and Pruning Algorithm for Classification Tree Design. *IEEE Trans. Pattern Anal. Mach. Intell.* **1991**, *13*, 163–174. [CrossRef]
16. Bohanec, M.; Bratko, I. Trading Accuracy for Simplicity in Decision Trees. *Mach. Learn.* **1994**, *15*, 223–250. [CrossRef]
17. Osei-Bryson, K.-M. Post-pruning in decision tree induction using multiple performance measures. *Comput. Oper. Res.* **2007**, *34*, 3331–3345. [CrossRef]
18. Fournier, D.; Crémilleux, B. A quality index for decision tree pruning. *Knowl.-Based Syst.* **2002**, *15*, 37–43. [CrossRef]
19. Barrientos, F.; Sainz, G. Knowledge-Based Systems Interpretable knowledge extraction from emergency call data based on fuzzy unsupervised decision tree. *Knowl.-Based Syst.* **2012**, *25*, 77–87. [CrossRef]
20. Wei, J.M.; Wang, S.Q.; You, J.P.; Wang, G.Y. RST in decision tree pruning. In Proceedings of the Fourth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD 2007), Haikou, China, 24–27 August 2007; Volume 3, pp. 213–217. [CrossRef]
21. Pawlak, Z. Rough sets. *Int. J. Comput. Inf. Sci.* **1982**, *11*, 341–356. [CrossRef]
22. Knoll, U.; Nakhaeizadeh, G.; Tausend, B. Cost-sensitive pruning of decision trees. In *European Conference on Machine Learning*; Springer: Berlin/Heidelberg, Germany, 1994; pp. 383–386.
23. Bradley, A.; Lovell, B. Cost-sensitive Decision Tree Pruning: Use of the ROC Curve. In Proceedings of the Eighth Australian Joint Conference on Artificial Intelligence, Canberra, Australia, 3–17 November 1995; pp. 1–8.
24. Ting, K.M. *Inducing Cost-Sensitive Trees via Instance Weighting*; Springer: Berlin/Heidelberg, Germany, 1998; pp. 139–147. [CrossRef]
25. Sweeney, L. k-Anonymity: A Model for Protecting Privacy. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.* **2002**, *10*, 557–570. [CrossRef]
26. Chew, Y.J.; Ooi, S.Y.; Wong, K.; Pang, Y.H. Privacy Preserving of IP Address through Truncation Method in Network-based Intrusion Detection System. In Proceedings of the 2019 8th International Conference on Software and Computer Application, Penang, Malaysia, 19–21 February 2019.
27. Witten, I.H.; Frank, E.; Hall, M.A.; Pal, C.J. *Data Mining: Practical Machine Learning Tools and Techniques*; Morgan Kaufmann: Burlington, MA, USA, 2016; ISBN 0128043571.
28. Perona, I.; Arbelaitz, O.; Gurrutxaga, I.; Martin, J.I.; Muguerza, J.; Perez, J.M. Generation of the Database Gurekddcup. 2016. Available online: https://addi.ehu.es/handle/10810/20608 (accessed on 17 December 2021).
29. Perona, I.; Gurrutxaga, I.; Arbelaitz, O.; Martín, J.I.; Muguerza, J.; Ma, J. Service-independent payload analysis to improve intrusion detection in network traffic. In Proceedings of the 7th Australasian Data Mining Conference, Glenelg, Australia, 27–28 November 2008; Volume 87, pp. 171–178.
30. Moustafa, N.; Slay, J. UNSW-NB15: A comprehensive data set for network intrusion detection systems. In Proceedings of the 2015 Military Communications and Information Systems Conference (MilCIS), Canberra, Australia, 10–12 November 2015; pp. 1–6. [CrossRef]
31. Ring, M.; Wunderlich, S.; Grüdl, D.; Landes, D.; Hotho, A. Flow-based benchmark data sets for intrusion detection. In Proceedings of the 16th European Conference on Cyber Warfare and Security, Dublin, Ireland, 29–30 June 2017; pp. 361–369.
32. 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]
33. Al Tobi, A.M.; Duncan, I. Improving Intrusion Detection Model Prediction by Threshold Adaptation. *Information* **2019**, *10*, 159. [CrossRef]
34. Chew, Y.J.; Lee, N.; Ooi, S.Y.; Wong, K.-S.; Pang, Y.H. Benchmarking full version of GureKDDCup, UNSW-NB15, and CIDDS-001 NIDS datasets using rolling-origin resampling. *Inf. Secur. J. Glob. Perspect.* **2021**, 1–22. [CrossRef]
35. Shetu, S.F.; Saifuzzaman, M.; Moon, N.N.; Nur, F.N. A survey of Botnet in cyber security. In Proceedings of the 2019 2nd International Conference on Intelligent Communication and Computational Techniques (ICCT), Jaipur, India, 28–29 September; IEEE: New York, NY, USA, 2019; pp. 174–177.