Future Generation Computer Systems 102 (2020) 292–306

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

Interactive three-dimensional visualization of network intrusion detection data for machine learning

Wei Zong *, Yang-Wai Chow *, Willy Susilo

Institute of Cybersecurity and Cryptology, School of Computing and Information Technology, University of Wollongong, Australia

HIGHLIGHTS

- Approach to visualizing network intrusion detection data in 3D for cybersecurity.
- Facilitates understanding of data via an interactive visual representation.
- Visualization of machine learning decision spaces.
- Visual observation and insight into machine learning misclassification.

ARTICLE INFO

Article history:
Received 14 December 2018
Received in revised form 9 May 2019
Accepted 21 July 2019
Available online 1 August 2019

Keywords:
Cybersecurity
Network intrusion detection
Machine learning
Visualization

ABSTRACT

The threat of cyber-attacks is on the rise in the digital world today. As such, effective cybersecurity solutions are becoming increasingly important for detecting and combating cyber-attacks. The use of machine learning techniques for network intrusion detection is a growing area of research, as these techniques can potentially provide a means for automating the detection of attacks and abnormal traffic patterns in real-time. However, misclassification is a common problem in machine learning for intrusion detection, and the improvement of machine learning models is hindered by a lack of insight into the reasons behind such misclassification. This paper presents an interactive method of visualizing network intrusion detection data in three-dimensions. The objective is to facilitate the understanding of network intrusion detection data using a visual representation to reflect the geometric relationship between various categories of network traffic. This interactive visual representation can potentially provide useful insight to aid the understanding of machine learning results. To demonstrate the usefulness of the proposed visualization approach, this paper presents results of experiments on commonly used network intrusion detection datasets.

1. Introduction

With the increasing threat of cyber-attacks in the digital world today, cybersecurity experts have undertaken wide ranging studies on techniques for combating such security threats. The use of Machine Learning (ML) techniques in Network Intrusion Detection Systems (NIDS) is a growing area of research. ML techniques have long been used for analyzing and extracting useful information from data [1]. As such, the adoption of ML techniques in NIDS has been researched extensively due to its potential to provide promising solutions for automating the real-time detection of attacks or abnormal traffic patterns in a network [2,3]. In light of this, it is vital to understand the significance and reasons behind intrusion detection results that are produced by ML models [3].

Misclassification is a common problem in ML, and a general lack of insight into why such misclassification occurs impedes the development and improvement of ML models. While many studies focusing on the use of ML in NIDS have presented improved results when comparing with other techniques, there is not much emphasis on understanding the reasons for the improved results or lower misclassification rates [4–6]. These studies tend to present numeric results in the form of tables or graphs to compare performances between different ML techniques. Nevertheless, the underlying reasons for poor performance in the detection of certain attack categories are not usually explained and cannot be perceived clearly or intuitively. Without a comprehensible approach to analyzing the reasons for poor performance, the improvement of ML models usually relies on a trial-and-error process due to the complex nature of ML mechanisms [7]. More specifically, ML heavily depends on the characteristics of the training and testing datasets. Hence, an intuitive and explainable analysis of such datasets can facilitate the understanding of ML detection results.

* Corresponding authors.
E-mail addresses: wz630@uowmail.edu.au (W. Zong), caseyc@uow.edu.au (Y.-W. Chow), wsusilo@uow.edu.au (W. Susilo).

https://doi.org/10.1016/j.future.2019.07.045
0167-739X/

Information visualization can play an important role in analyzing datasets. This is because dataset visualization can facilitate mental perception and provide insight into complex data structures. Furthermore, when compared with numeric data presentation, visual representation is more inspiring and intuitive [8]. Researchers have already investigated various forms of visualization for applications in network security, ranging from two-dimensional (2D) to three-dimensional (3D) visualization approaches [9,10]. In addition, research in network security visualization covers a variety of domains, including for monitoring network traffic characteristics [11,12] and for visualizing complex attack patterns [13–15]. While studies have performed analysis on NIDS datasets [16,17], there is currently not much research on visualization systems for providing visual representation of NIDS data, which can provide insight into understanding ML detection results in NIDS.

This paper presents an interactive 3D visualization approach for analyzing NIDS datasets and ML results. The aim of the proposed approach is to provide a visual representation of data from NIDS datasets in a manner that portrays the geometric relationship between diverse network traffic data records [18], and to create a way of examining the likely causes of ML misclassification from a visual perspective. The visualization system allows users to interactively navigate through the NIDS data using a virtual camera in real-time, and to visually examine ML classifier decision spaces as well as to observe boundaries where misclassification occurs. In addition, users can obtain statistical information by interactively selecting parts of the data and can also extract selected data from the visual representation into a text format for detailed inspection.

Since the purpose of the proposed approach is to use visualization to facilitate analysis of ML detection results, the same procedure as how ML models are commonly trained is used in this research. A ML model is trained from the training set and applied to the testing set. A visual representation depicting the classification results along with the ML classifier decision spaces are presented to the user to allow for visual examination of the data. To increase the performance of ML models and to visualize high dimensional data, dimensionality reduction is useful to extract a subset of the original features or to convert the original data into lower dimensional space [4]. Janarthanan et al. [19] showed that by removing irrelevant and redundant features from NIDS datasets, computation costs decrease and high detection rates are still maintained. Examples of commonly used dimensionality reduction techniques in ML include the Principal Component Analysis (PCA) and information gain techniques [4,20]. The proposed approach adopts these reduction techniques in conjunction with the Support Vector Machine (SVM) ML technique. This paper demonstrates and discusses results from a prototype system that was developed based on the visualization approach proposed in this research.

The rest of this paper is organized as follows. Background information and related work are discussed in Section 2. Details of the proposed approach are then described in Section 3. Section 4 demonstrates and discusses results obtained from experiments that were conducted by implementing the proposed visualization approach. Finally, Section 4.3 concludes the paper and discusses future work.

2. Background and related work

This section presents a background to topics that are related to the work in this research. The related topics of dimensionality reduction and ML for network intrusion detection are described, followed by related work on network intrusion visualization and a description of network intrusion detection datasets.

2.1. Dimensionality reduction and machine learning in network intrusion detection

The use of ML techniques for network intrusion detection is a growing area of research, as these techniques can potentially provide a means for automating the detection of attacks and abnormal traffic patterns in real-time. As such, there is much interest and diverse research in this area. This section describes the related topics of dimensionality reduction and ML.

When training a ML model, dimensionality reduction is an important step that is usually performed before the training. The traditional approach to feature selection is where security experts would rank the importance of features manually. However, it would be ideal if an automated method for selecting important features can be employed. PCA is an important tool that has been used in a variety of work, including for research on analyzing network traffic for NIDS. Moustafa et al. [4] adopted the PCA technique to reduce high dimensionality of network connections. Hoz Correa et al. [21] used PCA for dimensionality reduction and noise removal in NIDS, while Lakhina et al. [22] proposed to use PCA as a statistical tool in network anomaly detection. In their approach, network traffic data was divided into normal and abnormal subspaces, before statistical analysis was performed to detect anomalies.

In other work, Camacho et al. [23] introduced the main steps of the Multivariate Statistical Process Control approach for NIDS based on the PCA technique. They also suggested that before applying PCA, the importance of each feature on the classification process should be considered. The reason for this is because only certain important features in the data may be essential to the classification process. Information gain is another automated approach to dimensionality reduction. Using this approach, features with low information gain can be eliminated from the classification process. Features with low information gain are considered to be unimportant as they have relatively small relevance on classification. Information gain can only be calculated using discrete variables, and it is equal to subtracting the sum of entropy for each subset of records, weighted by their probability of occurring, from the entropy of the target feature of the original dataset. A method of calculating information gain is provided as follows [20].

Let X and Y be discrete variables representing sample attributes $(x_1, x_2, \ldots, x_m)$ and class attributes $(y_1, y_2, \ldots, y_n)$, respectively. Then, the information gain, *IGain*, of a given attribute X regarding a class attribute Y is calculated as:

$$IGain(Y, X) = Entropy(Y) - Entropy(Y|X) \quad (1)$$

where

- $Entropy(Y) = -\sum_{i=1}^{n} P(Y = y_i) log_2 P(Y = y_i)$, where $P(Y = y_i)$ is the probability that y_i occurs, and
- $Entropy(Y|X) = -\sum_{j=1}^{m} P(X = x_j) Entropy(Y|X = x_j)$.

In conjunction with dimensionality reduction, there are a variety of ML techniques that have been used for network intrusion detection, including Support Vector Machine (SVM), Random Forests, etc. [2,24]. For example, Lin et al. [6] proposed a novel feature representation method, which considers the geometric properties of datasets and uses the k-Nearest Neighbor (kNN) classifier to detect network attacks. Although their approach performs well in detecting normal traffic, DoS and probe attacks, the detection accuracy of their approach is not satisfactory for other categories like U2R and R2L attacks.

Wang et al. [5] proposed an approach that first used a fuzzy clustering technique to divide the training set into several subsets. Then different Artificial Neural Networks were trained on these subsets. Finally, a fuzzy aggregation module was used to

combine the detection results. Their experimental results demonstrated high performance on intrusion detection using this multi-stage approach. Moustafa et al. [4] propose a novel technique, called geometric area analysis based on trapezoidal area estimation for NIDS. While this approach is effective in detecting intrusions for both the NSL_KDD and UNSW-NB15 datasets, the reasons for misclassification are not really described in detail.

Many studies usually do not describe the reasons for ML misclassification. This is in part due to the incomprehensible nature of ML algorithms, where many users regard ML as a black box. Nevertheless, instead of using trial-and-error process, visualization has been proposed as a critical tool for understanding and improving ML models. This trend has attracted many researchers, which has resulted in various studies on the topic [7]. As an example, Rauber et al. [25] presented work to visualize relationships between learned representations of observations, and relationships between artificial neurons to give network designers highly valuable insight into how their systems operate. In addition, Liu et al. [7] proposed a visual analytics system which facilitates understanding, diagnosing and refining deep convolution neural networks.

2.2. Network intrusion visualization

There is a wide range of research on various visualization techniques. For instance, the t-distributed Stochastic Neighbor Embedding (t-SNE) technique is a well known visualization technique for machine learning. Laurens et al. [26] developed this non-linear dimensionality reduction technique to visualize data through a scatter plot in two or three dimensions. Their experiments showed that the t-SNE approach is able to show good clustering characteristics of various data types when used on a variety of datasets.

There are also a number of visualization approaches proposed in the NIDS domain. Ruan et al. [27] previously used PCA in conjunction with multi-dimensional scaling, for dimensionality reduction for visual analysis of the KDD99 dataset. Yelizarov et al. [13] presented a visual technique, which can be used to display the overall status of the network and to show complex attack patterns. A NIDS for contributing to situation awareness by helping users to understand the network security status and events, has also been proposed [14]. McKenna et al. [15] designed a cybersecurity dashboard to help network analysts in identifying and summarizing patterns within network data.

In other work, Onut et al. [28] projected network data in 3D space and try to distinguish different attacks in a visual manner. Examples of several attack scenarios were described and discussed in their work. A NIDS based on 2D visual presentation of network data was proposed by Corchado et al. [29]. In their work, network traffic was visualized based on 5 variables of each packet, namely, source port, destination port, size, timestamp and protocol. In their system, anomalies in network traffic showed up as different patterns compared to normal connections.

Despite the many efforts in the development of visualization methods in NIDS, visualization techniques aimed at facilitating the understanding of ML detection results for NIDS are scarce. The work in this paper attempts to address this.

2.3. Network intrusion detection datasets

Network intrusion detection datasets are essential for the development of NIDS, as well as for evaluating the effectiveness of various intrusion detection techniques. Benchmark datasets that are commonly used by the research community include the KDD98, KDD_CUP99 and NSL_KDD datasets. These datasets contain a number of different categories of network traffic including normal traffic and attacks, such as Denial of Service (DoS), probe, Remote to Local (R2L) and User to Root (U2R). However, it has been contended that these network intrusion detection datasets were generated more than a decade ago, and several studies have highlighted flaws in these datasets [16,30].

In light of this, the UNSW-NB15 dataset was proposed as a contemporary dataset that was created as a hybrid of modern normal and contemporary synthesized attack activities of network traffic [17]. This dataset breaks down the attack categories into other attacks like worms, shellcode, exploits, fuzzers, etc. While experiments conducted in this study focused on using the NSL_KDD [16] and the UNSW-NB15 [17] datasets, the proposed approach is a generic visualization method that can also be applied to other datasets. The characteristics along with the features contained within the datasets used in this work are described on their respective websites: NSL_KDD[1] and UNSW-NB15.[2]

3. Proposed approach

The proposed interactive 3D visualization approach for analyzing NIDS datasets for machine learning is presented in this section. The overall process used to visualize network traffic in NIDS datasets is described. This is followed by a description of the method used to visualize the machine learning decision spaces for the different categories of network traffic.

3.1. 3D visualization of network traffic

Fig. 1 provides an overall depiction of the stages involved in the proposed approach. The first stage involves the extraction of network traffic records from the dataset. In view of the fact that in each network intrusion detection dataset, minor categories like worm attacks (in the UNSW-NB15 dataset), and U2R and R2L attacks (in the NSL_KDD dataset), only occupy a small portion of the dataset, all the data for the minor categories are extracted from the training set. The data from the remaining major categories are randomly extracted until a certain predefined amount, e.g., 30% of the data. While the full dataset can be used, the random sampling of data from the major categories does not diminish the quality of visual information, but it is useful to reduce the visual clutter and the amount of required computation on the visualization system. Extraction is not performed on the testing set, as all the test data is used.

Instead of using random sampling, Liu et al. [31] have previously suggested using blue noise sampling as this approach has two advantages over random sampling. It was reported that blue noise sampling better reduces data clustering and preserves more outliers from the data. This is because the distribution in blue noise sampling contains samples that are randomly located but remain spatially uniform [32]. Therefore, in the context of spatially uniform distribution, the use of blue noise sampling produces better results over random sampling.

However, for the method described in this paper, random sampling was used instead of blue noise sampling. The reason for this is because the purpose of the proposed visualization approach is to preserve the overall distribution of the original data, in order to observe visual characteristics of the data. In view of the fact that blue noise sampling can significantly reduce data clustering, the resulting sampling will likely alter the data distribution of the original dataset. In addition, the visualization of decision space boundaries may not produce useful information about the original dataset because changing the data distribution

[1] https://www.unb.ca/cic/datasets/nsl.html.

[2] https://www.unsw.adfa.edu.au/unsw-canberra-cyber/cybersecurity/ADFA-NB15-Datasets/.

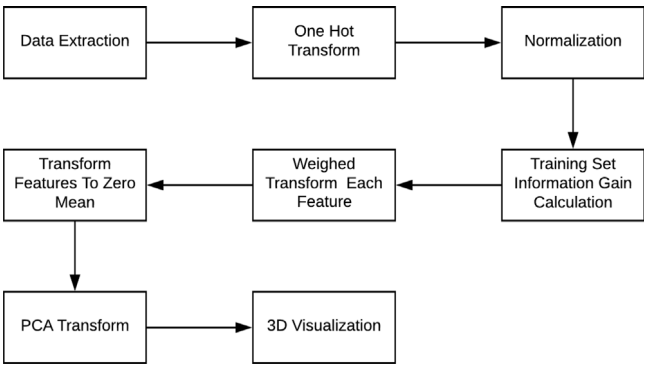


Fig. 1. Stages in the proposed approach.

will significantly affect the machine learning results. To overcome the limitations of random sampling on outliers, in the proposed approach minor categories of network traffic are not sampled, but rather all data for the minor categories are used. Random sampling is only performed on major categories to reduce visual clutter.

To visualize high dimensional data in 2D or 3D space, dimensionality reduction techniques must be applied to the data. There are a number of dimensionality reduction techniques that can potentially be used. For the purpose of the proposed visualization approach, the adopted dimensionality reduction technique must fulfill a number of requirements. First, it should preserve the geometric relationship between the network traffic records, since this relationship is important for ML models, like the SVM technique, in producing the ML decision spaces for the proposed approach, as described in Section 3.2 below. Second, it should learn a parametric mapping so that new data can be mapped into the low dimensional space. This requirement is essential because the proposed visualization process resembles the procedure for applying ML techniques, since the purpose of the approach is to facilitate the understanding of ML detection results. In particular, ML models are only trained using training sets, followed by testing on the testing sets. The last requirement is that there must be a way to inverse the data in low dimensional space back to high dimensional space, as this is used to determine the decision spaces (Section 3.2).

Hence, although popular dimensionality reduction methods like the t-SNE usually provides good data visualization results in other research work. It does not satisfy the requirements mentioned above and is thus not suitable for the proposed approach. t-SNE does not retain geometric relationship between data, does not learn a parametric mapping and the transformation to low dimension space cannot be inversed. In contrast, the PCA technique fulfills all these requirements. As such, the PCA technique was the method used in this research. It should be noted that other dimensionality reduction techniques can be used as long as it fulfills the above mentioned requirements.

The next stage of the process is to apply one-hot transformation. The reason for performing this is because the direct use of categorical data is not suitable for the PCA technique. One-hot transformation replaces a categorical feature with a set of binary features. Each binary feature indicates whether the original categorical feature of a record is a certain value or not. The number of binary features equals the number of different values in the categorical feature.

The transformation is done on the training and testing sets individually. This is because some categorical values may only exist in the testing set and these values should not be used when training the PCA model. More specifically, the purpose is to train the PCA model based on the training set only, and then apply the trained PCA model to the testing set. Handling training sets and testing sets separately is a common practice when applying ML models. Since the intention of the visualization approach is to facilitate analysis of ML detection results, the PCA model in the proposed approach is trained using the same procedure as how ML models are commonly trained. As a result, all the possible features will be created because some features may appear in the training set but not in the testing set. Features that only appear in the testing set will be ignored.

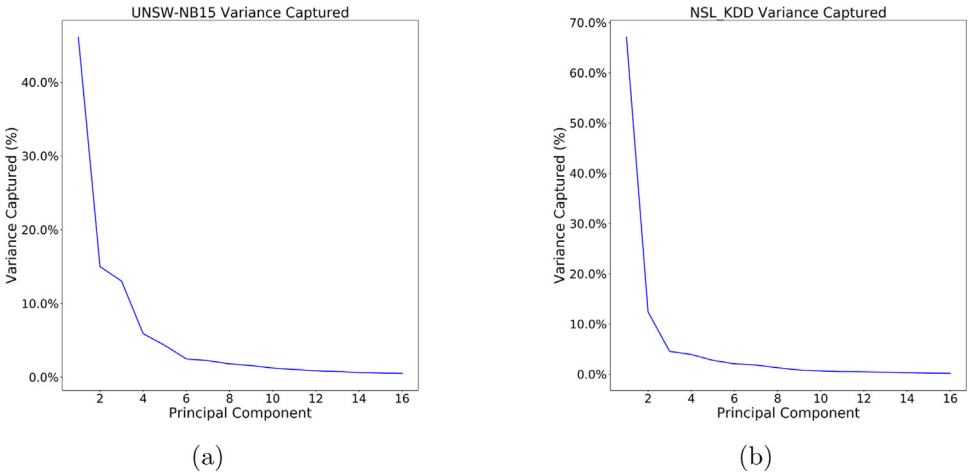
This is followed by a normalization process to normalize the range of the features. The purpose of performing normalization is because the range of values for the features can differ significantly. For example, some values may range between zero to less than a hundred, while others may range from 0 to several millions. To correctly utilize PCA, linear normalization is applied to each feature in both the training and testing sets to normalize the values to the minimum and maximum range of the training set. The detailed procedure is that every feature in the training set is divided by the difference of the maximum and minimum values of that feature. Then, all features in the testing set are also divided by the difference of the maximum and minimum values of that respective feature from the training set.

As previously mentioned, Camacho et al. [23] suggested that before applying PCA, the importance of each feature in relation to classification should be considered. Thus, the information gain of each feature is calculated to ascertain the importance of each feature of the training set. Since information gain requires discrete features, equal frequency binning discretization is performed. We use an adaptive method to choose a suitable frequency for each feature, since each feature may need a different frequency to discretize reasonably. For example, if the values of a feature are all different, we may need a big frequency value. Otherwise, it would result in too many categories and the information gain calculation would be ineffective. Therefore, we predefine a maximum value, 1000 in our experiment, for discretized categories. We first try a small value for the frequency, 10 in our experiment, and get the number of discretized categories. If this number does not exceed the maximum value, we accept the result of equal frequency binning discretization on this feature. Otherwise, we will double the value of the frequency and try discretization again until the number of discretized categories does not exceed the maximum value.

After the numeric features are discretized, we calculate the information gain values for each feature. Then, the information gain values are normalized between 0 and 1, and weighted transform is applied by multiplying each feature with the corresponding information gain value. As a result, the variances of important features either remain or are decreased slightly, whereas the variances of unimportant features are significantly reduced.

Then PCA is applied to all features in the datasets. To correctly apply PCA transformation, the mean of each feature must be zero. Hence, all features are transformed to zero mean in the training set, and the testing set is transformed based on mean values of the corresponding features from the training set. The PCA model is trained on the training set and then used to transform both the training and testing sets. The resulting variance of the first 16 principal components for the NSL_KDD and UNSW-NB15 datasets is shown in Figs. 2(a) and 2(b), respectively. It should be noted that the first 3 components capture a significant portion of variances, representing 74.1% of the variances for the UNSW-NB15 dataset and 84.0% for the NSL_KDD dataset.

Finally, the 3D coordinate of a network traffic record is computed as a weighted sum of its first 3 PCA components, c_1, c_2, c_3, with a normalized vector basis in 3D space: v_1, v_2, v_3. The (x, y, z) components of the 3 vectors we used are provided in Table 1.



(a) (b)

Fig. 2. Variance of the first 16 principal components; (a) for the UNSW-NB15 dataset; (b) for the NSL_KDD dataset.

Table 1
(x, y, z) components of a vector basis in 3D space.

	(x, y, z) components
v_1	(1, 0, 0)
v_2	(0, 1, 0)
v_3	(0, 0, 1)

The resulting 3D coordinate is then multiplied by a constant scaling factor, s, which is used to control the degree of separation between positions. The formula for calculating a 3D coordinate is as follows:

$$position = \sum_{i=1}^{3} c_i \hat{v}_i s \tag{2}$$

Spheres with different colors are then rendered at these positions to represent different categories of network traffic. It should be noted that the data can also be visualized in 2D space, if only the first two PCA components are used in conjunction with a normalized vector basis in 2D space. However, in view of the fact that the third component captures 13.0% and 4.5% variances for UNSW-NB15 and NSL_KDD datasets respectively, in this paper, we only focus on 3D visualization as the three components obviously capture more variance in both datasets.

The proposed approach works well on datasets where a significant portion of variances can be captured by a few components, because only the first two or three components are used for visualization. Otherwise, the visualization may result in a situation where no obvious clusters can be observed, because there may not be enough variance to differentiate between records.

3.2. Decision space

A voxel-based approach was adopted to visualize the ML decision space for different categories of network traffic. The purpose of this decision space is to provide a visual representation to depict the predicted 3D space in which a ML model will classify the categories of network traffic. For example, if a network traffic instance lies within a Denial of Service (DoS) attack decision space, the ML model will determine that network traffic instance is a DoS attack. On the other hand, if an instance of a DoS attack lies outside the decision space, it will be misclassified as some other category. In this manner, by presenting a 3D visualization of the decision space to a user, the user will be able to see whether a particular ML model can accurately classify the network traffic, as well as where misclassification occurs.

The commonly used Support Vector Machine (SVM) technique was used to demonstrate the effectiveness of the ML decision space. For this, an SVM using C-support vector classification was trained using the first 13 components in the training set. The trained SVM was then used to detect different categories in the testing set. Support for multi-category network traffic classification was performed using a one-vs-one scheme. The reason why the first 13 components were used is because these 13 components are the most significant features, representing 96.3% of the variances for the UNSW-NB15 dataset and 98.7% for the NSL KDD dataset. The remaining components only occupy a very small portion of the variances. It has previously been reported that omitting irrelevant and redundant features for training ML models can reduce the computation costs while maintaining high detection rates [19]. Hence, only the first 13 components were used in the proposed approach.

In our experiments to illustrate the voxel-based decision space, the 3D visualization space was divided into 150 voxels along each of the x, y and z axes respectively, resulting in a total of over 3 million voxels. Note that while increasing the number of voxels will improve the visualization resolution, it will also increase the overall computation time. The center position of each voxel, which is in 3D space, is extended to thirteen-dimensions (13Ds) by initializing the additional dimensions with zero values, then passing this as input to the SVM. The predicted network traffic category is considered to be the type of that voxel (e.g., a DoS voxel). The voxels were then rendered based on their respective type, thus creating a 3D visualization of the ML decision space. Since the additional dimensions are initialized to zeros to extend the 3D data to 13Ds, this may potentially result in inaccuracies in the decision boundaries in 3D space. An illustration of this is shown in the limitations section of the experiment results, i.e. Section 4.3.

4. Experiment results and discussion

This section demonstrates results obtained from the system that was developed based on the proposed visualization approach. The visualization prototype was developed using scikit-learn 0.19.1 [33] in conjunction with Unity 2017.3.1f1. The programming languages used were Python and C# respectively. For the hardware setup, the CPU was an Intel(R) Core(TM) i5-3317U 1.70 GHz (quad-core), with 4 GB of RAM and the graphics card was an NVIDIA GeForce GT 740M. For the experiments, the

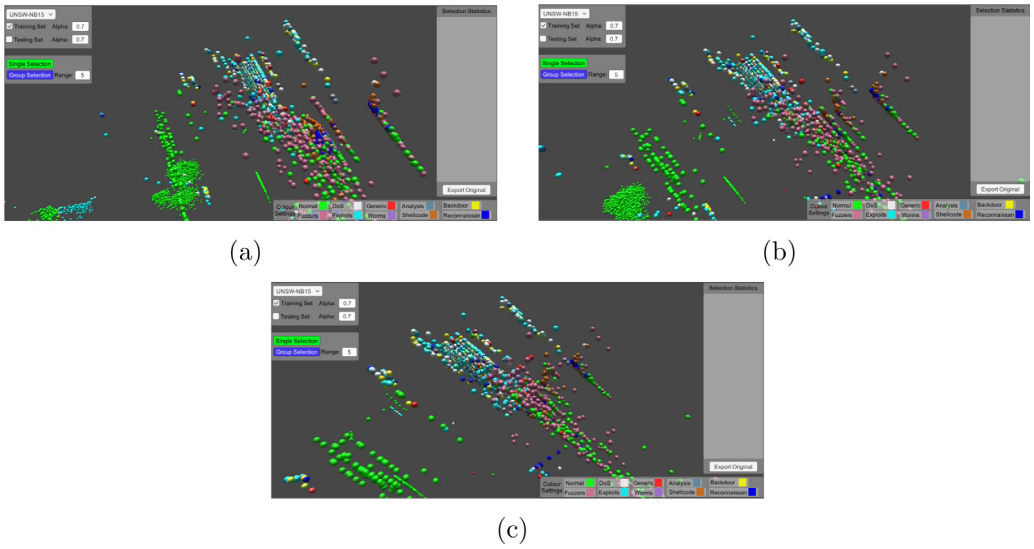


Fig. 3. Interactive virtual camera; (a) example of a screen capture from the virtual camera's viewpoint; (b) virtual camera's moved and rotated slightly; (c) virtual camera's moved and rotated more.

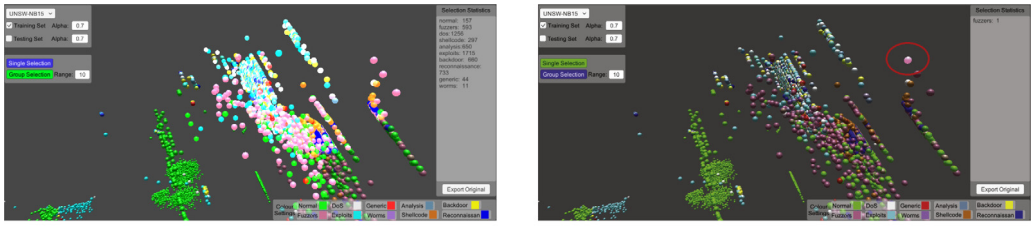


Fig. 4. Information on selected data; (a) selecting a group of data points; (b) selecting a single data point.

UNSW-NB15 and NSL_KDD datasets were used to illustrate the results, and some representative observations from these datasets are presented here.

It should be noted that the proposed 3D visualization system is an interactive system that is displayed in real-time. Users can freely navigate the virtual camera, with 6 degrees of freedom, in 3D space to examine different areas of the display and to observe ML decision boundaries. An example of this can be seen in the images shown in Fig. 3, where each image is a screen capture taken from different virtual camera positions and orientations. It can also be seen that the transparency of data points can be adjusted (using the 'Alpha' value in the upper left corner) to be able to better see overlapping data points. Furthermore, users can interactively select portions of the data to obtain statistical information, as well as extract selected data from the visual representation into a text format for closer inspection. Fig. 4(a) demonstrates how users can select a group of data points within an adjustable radius around a data point to obtain information about the group, whereas, Fig. 4(b) shows a single selected data point. Selected data points are highlighted in a brighter color. Once the data is selected, users can extract detailed information on the selected data into a text file.

4.1. UNSW-NB15 dataset

A 3D visual representation of the UNSW-NB15 dataset as displayed by the system is shown in Fig. 5. Fig. 5(a) shows a visual depiction of a portion of the training set from two different camera viewpoints, where the different network traffic categories have been color coded to visually distinguish them from one another. It can be seen visually that traffic from the same category are typically clustered together. The visual representation of part of the testing set from two different camera viewpoints is shown in Fig. 5(b). When comparing the visual representation of the training set with the testing set, it can clearly be seen that the characteristics of both sets are similar. This means that due to the similar patterns, when training a ML model using the training set it is highly likely that the model will be able to detect attacks in the testing set.

Fig. 5(c) provides an empirical grouping of data from the training set. The clusters circled in yellow depict sections that mainly only contain normal network traffic, whereas other clusters are circled in red. It can be seen from the groupings that the traffic for generic attacks are well clustered together. However, the traffic in the mixed clusters is not so well defined as they contain both attacks and normal network traffic.

Fig. 6 provides examples of certain clusters for closer examination. Examples of normal traffic clusters from the training and testing data are shown in Figs. 6(a) and 6(b), respectively. It can be seen from the figures that the characteristics of the training and testing data are very similar. The implication of this for ML is that traffic in clusters that contain nearly homogeneous records are easier to correctly identify and typically result in high detection performance. This is because their characteristics have similar patterns with little variation between them, which results in them being clustered together. Figs. 6(c) and 6(d) show examples of a cluster from the training and testing data, respectively, that contain mostly generic attacks. Note that the statistics of the traffic in the cluster is shown in the panel on the right. Since the cluster contains mainly homogeneous records, it can be

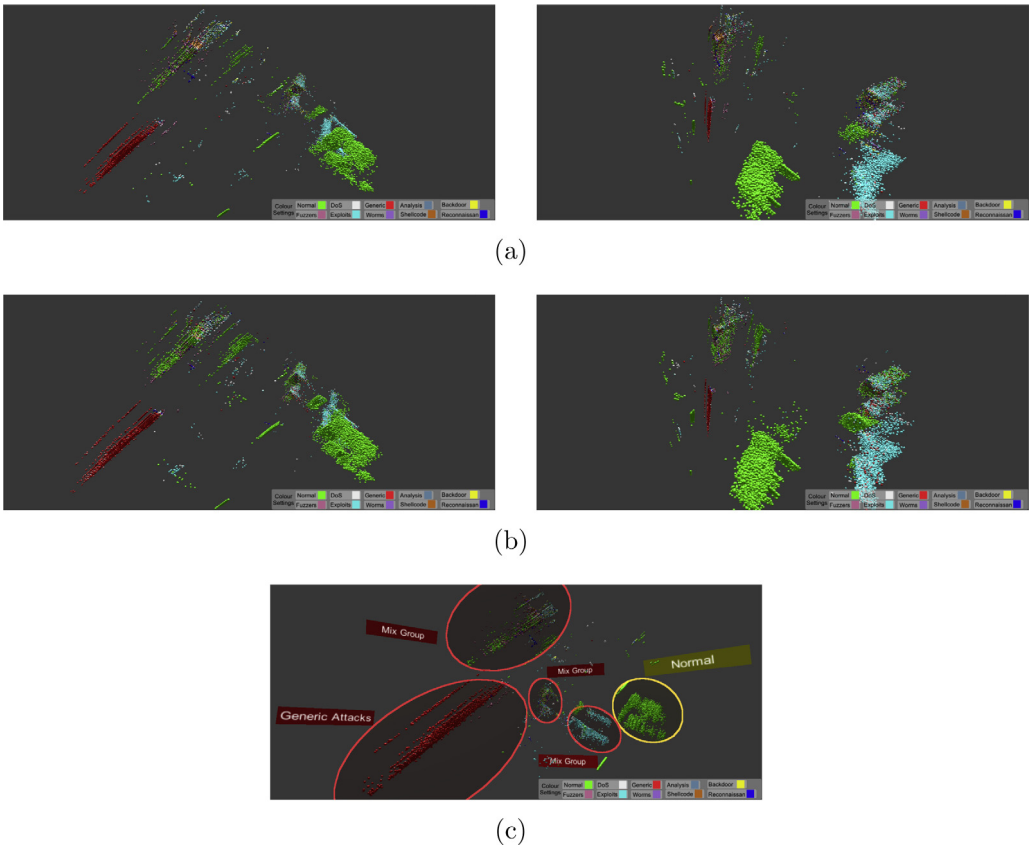


Fig. 5. 3D visual representation of the UNSW-NB15 dataset; (a) data from the training set from two different camera viewpoints; (b) data from the testing set from two different camera viewpoints; (c) empirical grouping of data from the training set.

concluded that in general the detection performance of most ML models on these generic attacks will be high. The results from several studies appear to support this conclusion [4,19].

However, in other sections of the dataset, the traffic is mixed together and there is no clear visual distinction between the different traffic categories within these clusters. An example of such a cluster is shown in Figs. 6(e) and 6(f) for the training and testing data, respectively. Such sections of the dataset are where ML techniques tend to face difficulties when it comes to the task of correctly identifying the individual categories of the traffic. In addition, the figures provide a visual comparison of the data between the training and testing set. There is a low number of worm attacks, which is disproportional when comparing the numbers of different traffic in the training set and the testing set. Furthermore, as there is a significant number of exploit attacks within the cluster, it is challenging for ML models to avoid misclassifying other traffic in the cluster as exploits.

Therefore, to address the problem of highly mixed clusters, a feature representation approach like the cluster center and nearest neighbor (CANN) method [6], can potentially be used to transform these records before the ML model is trained and also before using the trained model for detection. In view of the fact that the training and testing sets exhibit similar patterns, as can be seen from Figs. 5(a) and 5(b), it can be anticipated that if a feature representation approach successfully transforms the training data into clusters that contain mainly homogeneous records, the same approach should be able to transform the testing records into corresponding homogeneous clusters.

Experiments were also conducted to visualize the 3D ML decision space. The approach that was described in Section 3.2 was implemented, where an SVM using C-support vector classification was trained using the 13D training data and subsequently used to detect attacks in the testing set. The SVM that was used in the experiments was implemented by Pedregosa et al. [33]. The default parameters were used without performing any optimization, other values include: $gamma = 0.5$ and $C = 0.8$.

Table 2
Binary classification confusion matrix for the UNSW-NB15 dataset.

Predicted				
		Normal	Abnormal	Recall (%)
Actual	Normal	20 564	16 436	55.6
	Abnormal	203	45 129	99.6
	Precision (%)	99.0	73.3	

To demonstrate results of the 3D ML decision space, we first illustrate the approach using binary classification. In binary classification, network traffic instances were either classified as normal traffic or abnormal traffic. A confusion matrix of the results for binary classification is shown in Table 2, and visualization of the binary classification decision space of the trained SVM is depicted in Fig. 7. Fig. 7(a) shows an overview of the binary classification decision space, whereas Figs. 7(b) and 7(c) depict the decision space for normal traffic and abnormal traffic respectively. Network traffic instances that appear within a decision space should be classified based on the type of that space. For example, network traffic instances that are within the normal traffic decision space should be classified as being normal traffic.

From Table 2, it can be seen that while the recall rate of abnormal traffic is high, almost half of normal traffic was misclassified as abnormal traffic. The reason for this is because there are clusters that contain a mixture of both normal and abnormal traffic instances, in which the ML model cannot distinguish between them. The interactive visualization approach allows for a visual

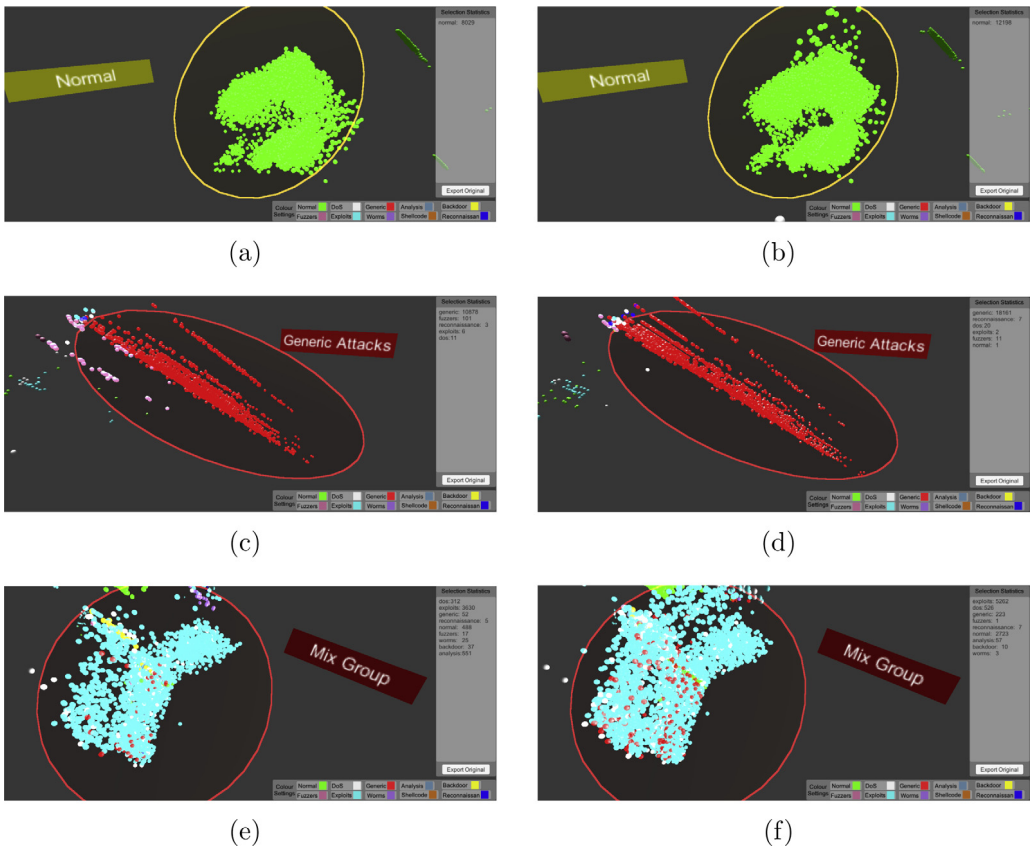


Fig. 6. Example clusters from the UNSW-NB15 dataset; (a)–(b) a cluster containing normal traffic from the training and testing data, respectively; (c)–(d) a cluster containing mostly generic attacks from the training and testing data, respectively; (e)–(f) a cluster containing mixed traffic from the training and testing data, respectively.

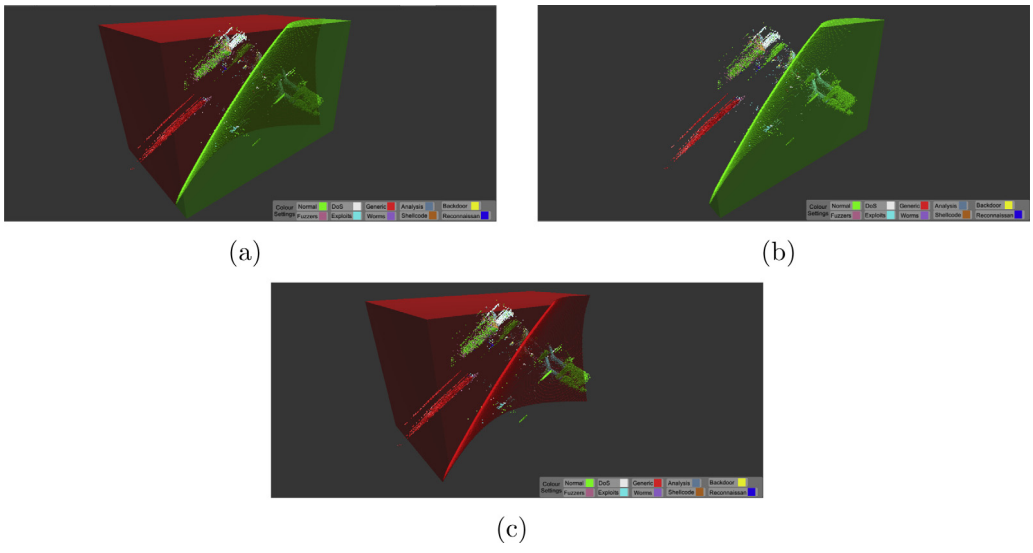


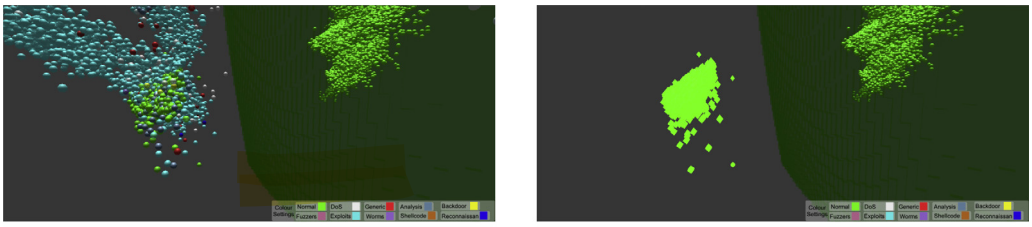
Fig. 7. Binary classification decision space for the UNSW-NB15 dataset; (a) overview of the decision space; (b) decision space for normal traffic; (c) decision space for abnormal traffic.

inspection of this. Fig. 8 presents an example of a close-up of the visualization that clearly shows where this misclassification occurs. In Fig. 8(a), the normal traffic decision space can be seen on the right of the figure and all normal traffic instances within that space should have been correctly classified. However, to the left of the figure there is a cluster of network traffic instances that are located outside the normal traffic decision space, which has a mixture of both normal and abnormal traffic. In Fig. 8(b), all abnormal traffic instances (from Fig. 8(a)) have been removed and the figure highlights the instances of normal traffic that the ML model misclassified as abnormal traffic. It can clearly seen that a large number of normal traffic instances have been misclassified as abnormal traffic.

Experiments were also conducted on visualizing the decision spaces for multi-category classification. Instead of binary

Table 3
Multi-category classification confusion matrix for the UNSW-NB15 dataset.

		Predicted										
		Analysis	Backdoor	DoS	Exploits	Fuzzers	Generic	Normal	Reconnaissance	Shellcode	Worms	Recall (%)
Actual	Analysis	0	0	0	619	0	0	58	0	0	0	0
	Backdoor	0	0	0	531	40	0	11	1	0	0	0
	DoS	0	0	0	3543	332	23	150	41	0	0	0
	Exploits	0	0	0	8979	1291	6	828	28	0	0	80.7
	Fuzzers	0	0	0	1769	3865	8	130	290	0	0	63.8
	Generic	0	0	0	389	261	18 161	41	19	0	0	96.2
	Normal	0	0	0	2844	10 766	1	21 985	1404	0	0	59.4
	Reconnaissance	0	0	0	824	1412	7	1	1252	0	0	35.8
	Shellcode	0	0	0	0	244	0	0	134	0	0	0
	Worms	0	0	0	34	8	0	0	2	0	0	0
	Precision (%)	0	0	0	46.0	21.2	99.8	94.7	39.5	0	0	



(a) (b)

Fig. 8. Close-up showing the decision space boundary and instances of misclassified traffic; (a) mixture of normal and abnormal traffic outside the decision space; (b) normal traffic misclassified as abnormal traffic.

classification where network traffic instances were merely classified as normal or abnormal, in multi-category classification, the ML model was used to classify all categories (e.g., normal, DoS, worm, exploits, etc.). A confusion matrix depicting the results for multi-category classification is shown in Table 3. Examples of decision spaces for the different categories are shown in Fig. 9. Fig. 9(a) provides an overview which shows all the different decision spaces. Fig. 9(b) to (g) show the decision spaces of other categories separately; namely, normal traffic, fuzzers, exploits, generic attacks, reconnaissance and DoS traffic, respectively.

From visual inspection of the 3D visualization results, it could clearly be observed that almost all the generic attack instances were contained within the generic attack decision space. This was confirmed in the high detection rate of generic attacks as shown in the confusion matrix in Table 3, with a recall of 96.2% and a precision of 99.8%. It should be noted that the decision spaces for reconnaissance traffic and DoS attacks, shown in Fig. 9(f) and (g) respectively, are comparatively small. As such, the figures show close-up screen captures of those decision spaces. Decision spaces for some categories like analysis and backdoor traffic did not appear in the visualization, because the basic SVM model used in the experiments was not able detect these categories of attacks.

Furthermore, as there was a greater proportion of exploit attacks than other categories in some clusters, it was anticipated that this would potentially result in other traffic being misclassified as exploits. This was verified in the results, as can be seen in the example shown in Fig. 10. Fig. 10(a) are close-ups of part of the exploits decision space from two different camera viewpoints; they show a mixture of exploits and other traffic that lie within the decision space for exploits. In Fig. 10(b), the traffic instances that were misclassified as exploits are highlighted from two different camera viewpoints.

4.2. NSL_KDD dataset

Fig. 11 shows the overall 3D visual representation of the NSL_KDD dataset as displayed by the proposed system. Fig. 11(a) depicts a portion of the data from the training set, while Fig. 11(b) displays part of the data from the testing set, from two different camera viewpoints respectively. It can be seen that while the overall visual distributions share relatively similar characteristics, the training and testing sets in the NSL_KDD dataset have more differences between them as compared with the UNSW-NB15 dataset (i.e. the characteristics of traffic between the training and testing sets in the UNSW-NB15 dataset exhibited higher similarities). Fig. 11(c) in turn shows the empirical grouping of data from the training set. Like the UNSW-NB15 dataset, it also shows that the data can be grouped into several clusters.

From the visual representation, one can observe that the data consists of clusters that contain mainly homogeneous records as well as clusters that contain diverse traffic. Figs. 12(a) and 12(b) show examples of DoS attack traffic from the training and testing data, respectively, from two different camera viewpoints. It can clearly be seen that the display of traffic within the cluster is isolated away from the rest of the traffic, and contains mostly homogeneous DoS attack records. Due to the isolation and clustering of such homogeneous data, ML techniques are expected to perform well when identifying such traffic.

Similarly, another example of a cluster containing distinguishable homogeneous traffic is one containing probe attacks, as shown in Figs. 12(c) and 12(d) from the training and testing data, respectively. Although in Fig. 12(d) there are other traffic near the cluster, such as normal traffic, the majority of traffic inside the cluster is still probe attacks. However, ML models may misclassify normal traffic and other attacks near this cluster as probe attacks when applied to the testing set.

Unlike the previous two examples, the main difficulty faced by ML techniques in the NSL_KDD comes from previously unknown attacks in the testing set [34]. Even though the categories of attacks are the same in the training and testing sets, the characteristics of these traffic significantly differ. A visual representation of this comparing the training and testing data is shown in Figs. 12(e) and 12(f), respectively. It can be seen that in the training set in Fig. 12(e) that there are almost no R2L attacks within this cluster. On the other hand, there are many R2L attacks in the corresponding cluster from the testing set, as shown in

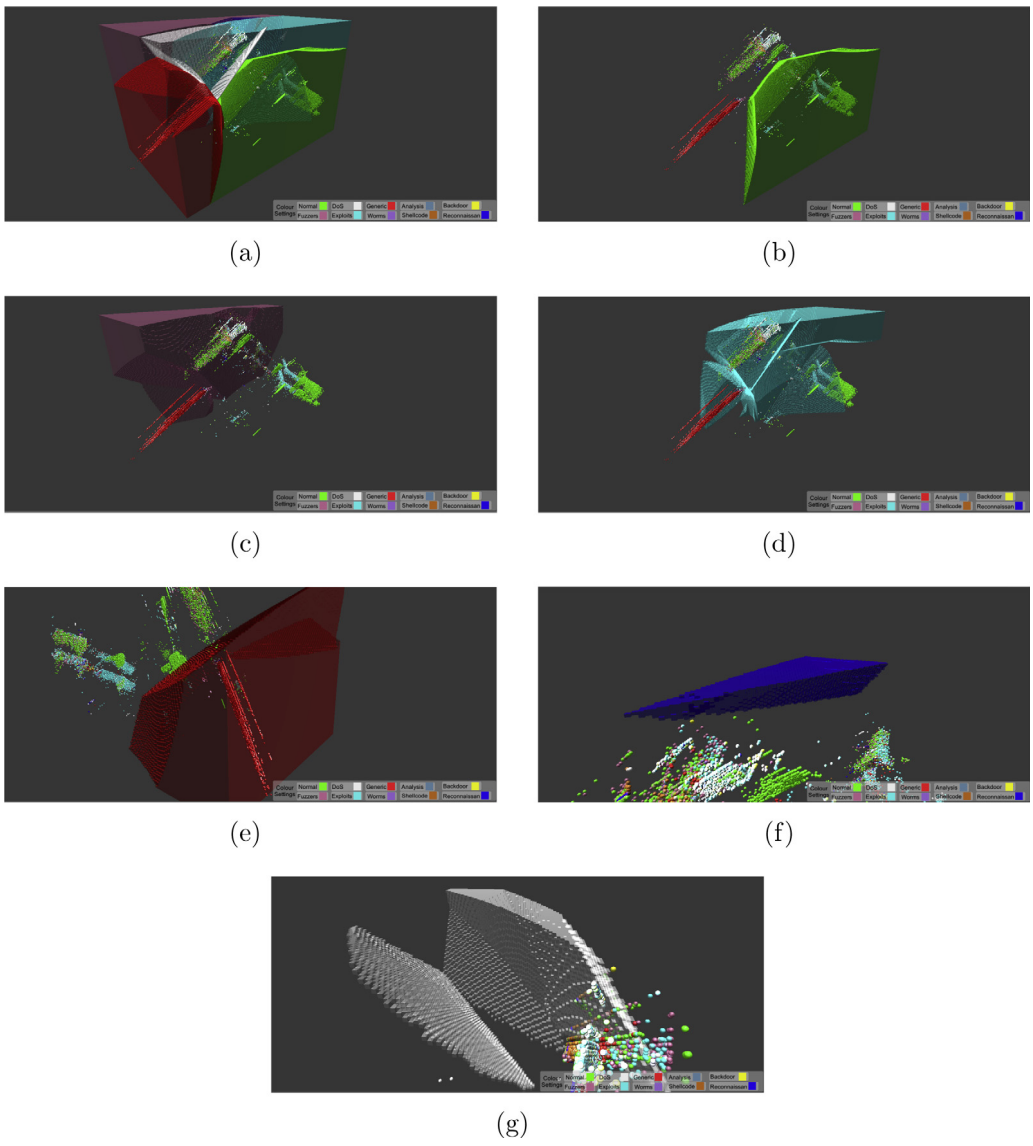


Fig. 9. Multi-category classification decision spaces for the UNSW-NB15 dataset; (a) overview of the decision spaces; (b) decision space for normal traffic; (c) decision space for fuzzers; (d) decision space for exploits; (e) decision space for generic attacks; (f) decision space for reconnaissance traffic; (g) decision space for DoS traffic.

Fig. 12(f). Despite the fact that ML models should be designed based on the training set, ML techniques that do not consider such abnormal characteristics may perform unsatisfactorily when it comes to detecting such attacks.

For visualizing the decision space, the same set of experiments that were conducted on the UNSW-NB15 dataset were performed on the NSL_KDD dataset. Table 4 shows the confusion matrix for binary classification, and the decision space visualization for binary classification is portrayed in Fig. 13. Fig. 13(a) shows an overview of the binary decision space, whereas Figs. 13(b) and 13(c) show the decision spaces for normal and abnormal traffic, respectively. Fig. 13(d) illustrates an example of abnormal traffic instances that are correctly classified and they are outside the normal traffic decision space.

For results of multi-category classification, a confusion matrix of the results is shown in Table 5 and the images of the decision spaces are shown in Fig. 14. Fig. 14(a) gives a overall depiction of the decision spaces. Figs. 14(b) to 14(d) shows the individual decision spaces for normal traffic, DoS attacks and probe attacks, respectively. It should be mentioned that although there is no R2L decision space in 3D space, some R2L attacks can still be correctly classified in 13D space as shown in Table 5.

Table 4
Binary classification confusion matrix for the NSL_KDD dataset.

		Predicted		
		Normal	Abnormal	Recall (%)
Actual	Normal	8918	792	91.8
	Abnormal	5208	7625	59.4
	Precision (%)	63.1	90.6	

Table 5
Multi-category classification confusion matrix for the NSL_KDD dataset.

		Predicted					
		DoS	Normal	Probe	R2L	U2R	Recall (%)
Actual	DoS	5378	2018	64	0	0	72.1
	Normal	473	8959	271	7	0	92.3
	Probe	162	844	1415	0	0	58.4
	R2L	2	2548	322	13	0	0.5
	U2R	0	59	4	4	0	0.0
	Precision (%)	89.4	62.1	68.2	54.2	0.0	

As was previously discussed, ML models perform well on clusters that contain homogeneous traffic, but do not perform well when it comes to unknown attacks. Figs. 15(a) and 15(b)

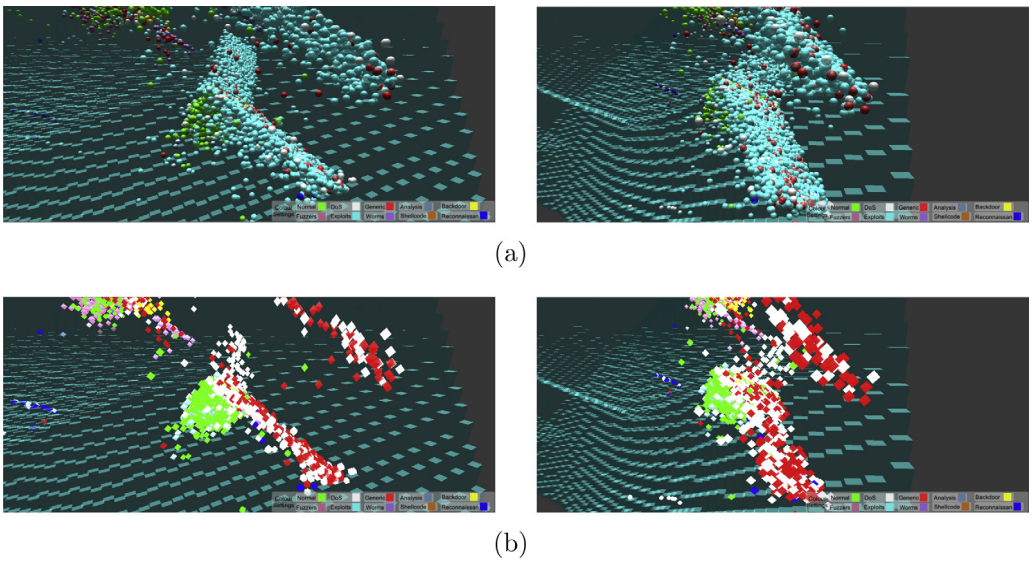


Fig. 10. Close-up showing a part of the decision space for exploits and instances of misclassified traffic; (a) mixture of exploits and other traffic inside the exploits decision space from two different camera viewpoints; (b) traffic that was misclassified as exploits from two different camera viewpoints.

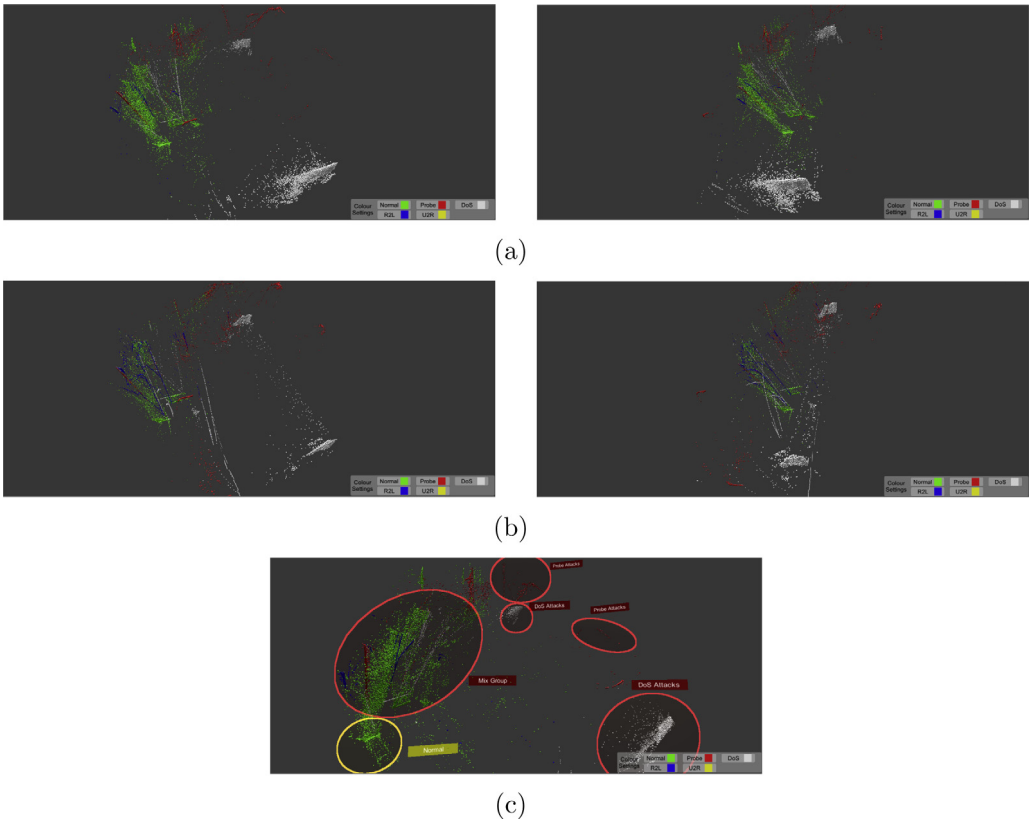


Fig. 11. 3D visual representation of the NSL_KDD dataset; (a) data from the training set from two different camera viewpoints; (b) data from the testing set from two different camera viewpoints; (c) empirical grouping of data from the training set.

show examples of a section of the DoS decision space for the training and testing data, respectively, where the DoS attacks are correctly classified due to the isolation and clustering of such homogeneous data in both the training and testing data. On the other hand, since the ML model is trained using the training data, if the training data does not adequately represent the network traffic, this can affect the detection accuracy of the ML model. As an example, Fig. 16(a) shows part of the normal traffic decision space from the training data, whereas Fig. 16(b) shows part of the normal traffic decision space from the testing data. It can be seen in Fig. 16(b) that R2L attacks are within the normal traffic decision space in the testing set and they have been misclassified as normal traffic. The reason for this is because these attacks were previously unknown in the training data. In addition, there are a number of DoS attacks in both Figs. 16(a) and 16(b). However, while samples are present in the training set, they are misclassified because they only occupy a small portion of that region in the training set.

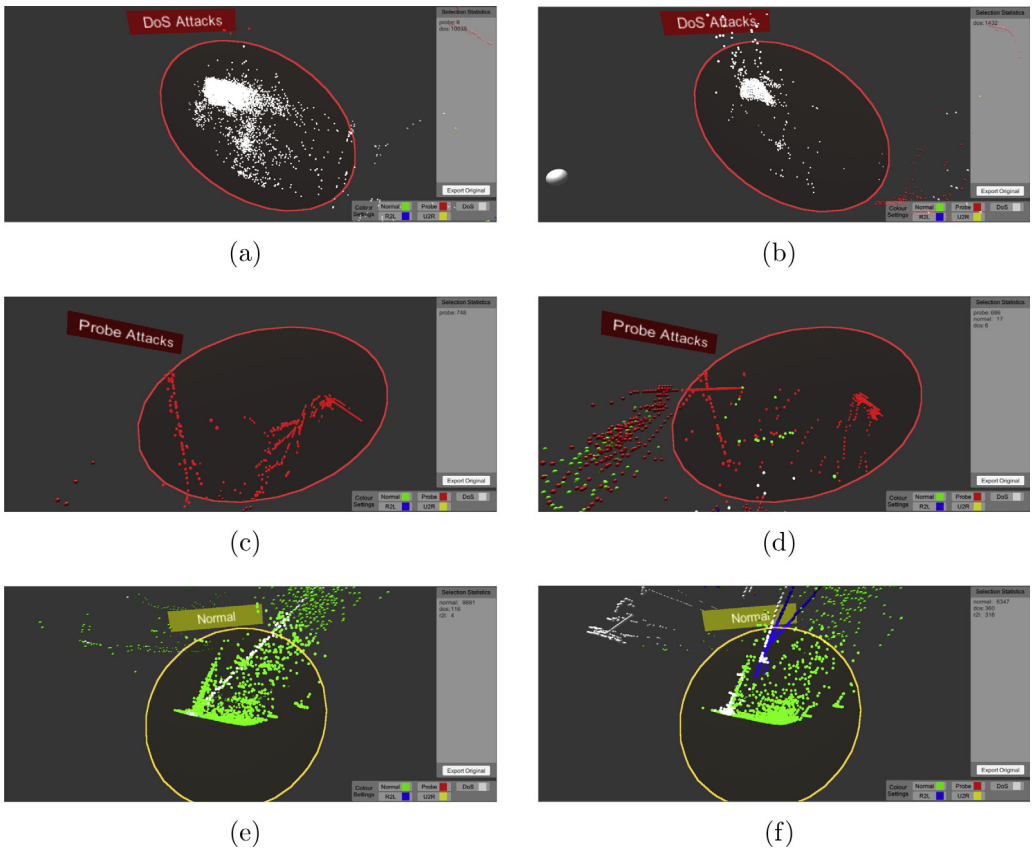


Fig. 12. Examples of clusters from the NSL_KDD dataset; (a)–(b) a cluster containing DoS attacks from the training and testing data, respectively; (c)–(d) a cluster containing mostly probe attacks from the training and testing data, respectively; (e)–(f) a cluster containing mainly normal traffic from the training and testing data, respectively.

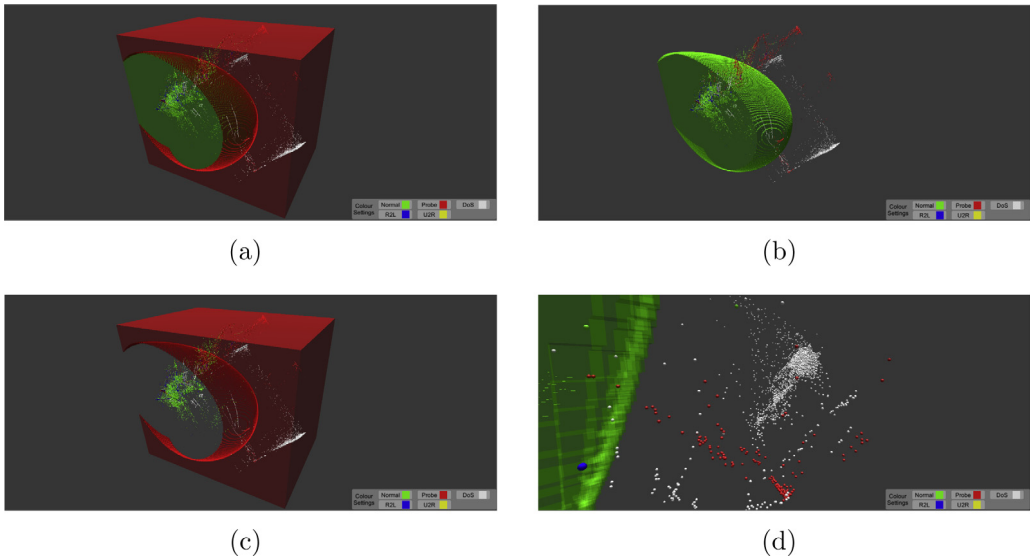


Fig. 13. Binary classification decision space for the NSL_KDD dataset; (a) overview of the decision space; (b) decision space for normal traffic; (c) decision space for abnormal traffic; (d) abnormal traffic instances located outside the normal traffic decision space.

4.3. Limitation and future work

Fig. 17 illustrates an example of a limitation with the visualization approach. In the figure, only normal traffic and the decision space of normal traffic is shown, with misclassified normal traffic highlighted using the diamond shapes. It is expected that all normal traffic that lie outside the normal traffic decision space should be misclassified as abnormal traffic. However, as can be seen within the red circle, there are normal traffic that were correctly classified, even though they are located outside the normal traffic decision space.

The cause of this is the loss of information when the decision space voxel positions, which are in 3D space, were inversed back to 13Ds as input to the SVM. In other words, the decision space is

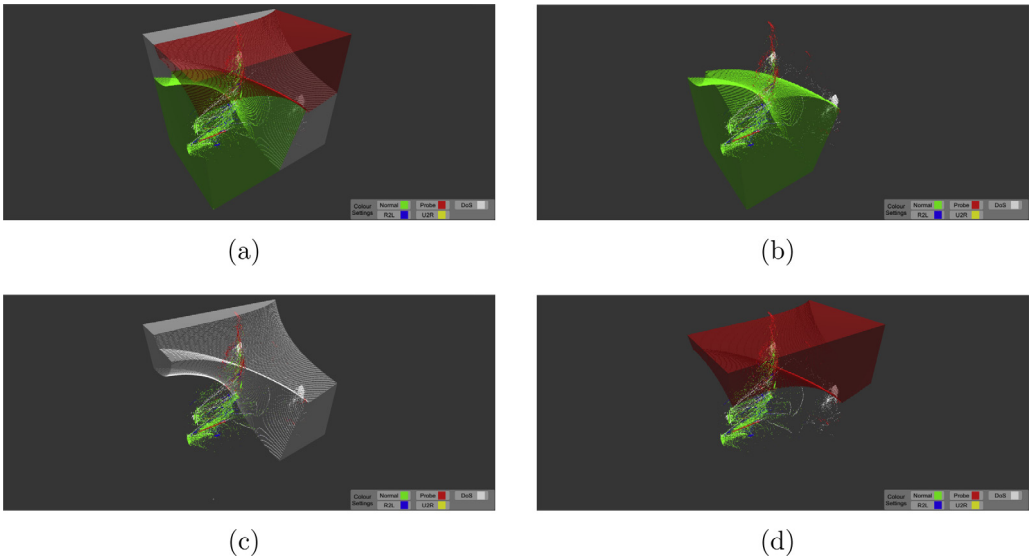


Fig. 14. Multi-category classification decision spaces for the NSL_KDD dataset; (a) overview of the decision spaces; (b) decision space for normal traffic; (c) decision space for DoS attacks; (d) decision space for probe attacks.

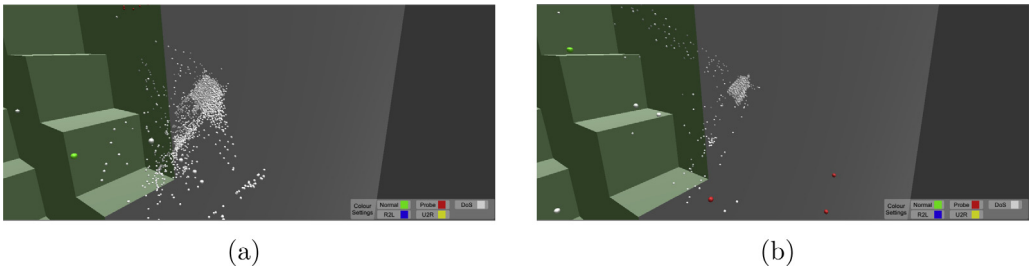


Fig. 15. Close-ups showing a section of the DoS decision space for the training and testing data; (a) traffic from the training data; (b) traffic from the testing data.

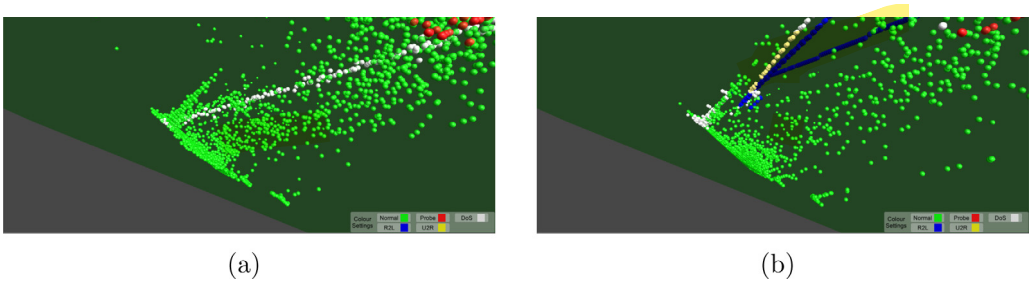


Fig. 16. Close-ups showing misclassification of attacks not represented adequately in the training data; (a) traffic instances in the training data; (b) misclassified traffic in the testing data.

not absolutely accurate as can be seen in Fig. 17. Nevertheless, in general the proposed decision space visualization approach can still provide useful insight when examining visual patterns in the ML detection results, as discussed in the preceding sections.

The visualization approach works well on UNSW-NB15 and NSL_KDD datasets, because a significant portion of variances in the data are captured by first few principal components, as shown in Fig. 2. Otherwise, the visualization may result in a situation where no obvious clusters can be observed as there may not be enough variance to differentiate between network traffic instances.

The experiments conducted in this study show different geometric characteristics in the visual representation of the UNSW-NB15 and NSL_KDD datasets. It was highlighted that the main challenge presented to ML techniques in the UNSW-NB15 dataset is due to sections that contain clusters with highly heterogeneous data, whereas the main challenge in the NSL_KDD dataset is due to the abnormal characteristics of unknown attacks. This demonstrates the usefulness of the proposed 3D visualization approach in identifying patterns and clusters in the data. It was suggested that to address the problem of highly mixed clusters,

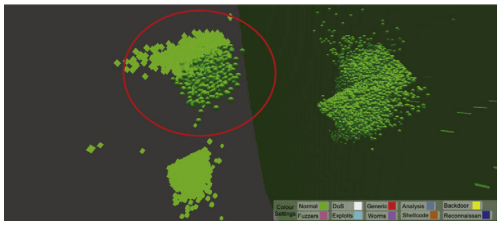


Fig. 17. Inaccuracies due to dimension reduction.

a feature representation approach like the cluster center and nearest neighbor (CANN) method [6], may potentially be used to transform network traffic records before the ML model is trained. This will be investigated in future work, along with the use of other dimensionality reduction and ML techniques, in conjunction with the visualization approach proposed in this research.

The current system deals with data from a network intrusion detection dataset, the end goal is for the system to be able to handle live network traffic. This will allow cybersecurity experts to examine and recognize patterns of incoming network traffic in real-time. In addition, since the visualization approach in this work is a general approach that can be used for visualizing different datasets and ML decision space results, it can be used in domains other than for network intrusion detection.

5. Conclusion

This paper presents an approach to visualizing network intrusion detection data in 3D. The aim of the proposed approach is to facilitate the understanding of NIDS datasets using a visual representation to reflect the geometric relationship and ML decision spaces between various categories of network traffic. This can provide useful insight for understanding ML detection results in NIDS, such as to understand the reasons for high misclassification rates in certain ML models.

A system was developed based on the proposed visualization approach, and results of experiments on commonly used NIDS datasets were presented. This demonstrates the usefulness of the proposed 3D visualization approach in identifying patterns and clusters in the data. Future work will focus on using other dimensionality reduction and machine learning techniques, and also on the visualization of live network traffic to allow users to examine incoming network traffic in a real-time interactive manner.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This work was supported by the NSW Cybersecurity Network and the NUW Alliance research grants.

References

[1] S. Suthaharan, Big data classification: problems and challenges in network intrusion prediction with machine learning, SIGMETRICS Perform. Eval. Rev. 41 (4) (2014) 70–73, http://dx.doi.org/10.1145/2627534.2627557, URL http://doi.acm.org/10.1145/2627534.2627557.

[2] A.L. Buczak, E. Guven, A survey of data mining and machine learning methods for cyber security intrusion detection, IEEE Commun. Surv. Tutor. 18 (2) (2016) 1153–1176, http://dx.doi.org/10.1109/COMST.2015.2494502.

[3] R. Sommer, V. Paxson, Outside the closed world: On using machine learning for network intrusion detection, in: 31st IEEE Symposium on Security and Privacy, S&P 2010, 16-19 May 2010, Berleley/Oakland, California, USA, IEEE Computer Society, 2010, pp. 305–316, http://dx.doi.org/10.1109/SP.2010.25.

[4] N. Moustafa, J. Slay, G. Creech, Novel geometric area analysis technique for anomaly detection using trapezoidal area estimation on large-scale networks, IEEE Trans. Big Data 1, http://dx.doi.org/10.1109/TBDATA.2017.2715166 URL doi.ieeecomputersociety.org/10.1109/TBDATA.2017.2715166.

[5] G. Wang, J. Hao, J. Ma, L. Huang, A new approach to intrusion detection using artificial neural networks and fuzzy clustering, Expert Syst. Appl. 37 (9) (2010) 6225–6232, http://dx.doi.org/10.1016/j.eswa.2010.02.102.

[6] W. Lin, S. Ke, C. Tsai, CANN: an intrusion detection system based on combining cluster centers and nearest neighbors, Knowl.-Based Syst. 78 (2015) 13–21, http://dx.doi.org/10.1016/j.knosys.2015.01.009.

[7] S. Liu, X. Wang, M. Liu, J. Zhu, Towards better analysis of machine learning models: A visual analytics perspective, Vis. Inform. 1 (1) (2017) 48–56, http://dx.doi.org/10.1016/j.visinf.2017.01.006.

[8] S. Liu, W. Cui, Y. Wu, M. Liu, A survey on information visualization: recent advances and challenges, Vis. Comput. 30 (12) (2014) 1373–1393, http://dx.doi.org/10.1007/s00371-013-0892-3.

[9] H. Shiravi, A. Shiravi, A.A. Ghorbani, A survey of visualization systems for network security, IEEE Trans. Vis. Comput. Graph. 18 (8) (2012) 1313–1329, http://dx.doi.org/10.1109/TVCG.2011.144.

[10] D. Staheli, T. Yu, R.J. Crouser, S. Damodaran, K. Nam, B.D. O'Gwynn, S. McKenna, L. Harrison, Visualization evaluation for cyber security: trends and future directions, in: K. Whitley, S. Engle, L. Harrison, F. Fischer, N. Prigent (Eds.), Proceedings of the Eleventh Workshop on Visualization for Cyber Security, Paris, France, November 10, 2014, ACM, 2014, pp. 49–56, http://dx.doi.org/10.1145/2671491.2671492, URL http://doi.acm.org/10.1145/2671491.2671492.

[11] R. Ball, G.A. Fink, C. North, Home-centric visualization of network traffic for security administration, in: C.E. Brodley, P. Chan, R. Lippmann, W. Yurcik (Eds.), Workshop on Visualization and Data Mining for Computer Security (VizSEC/DMSEC 2004), 29 October 2004, Washington DC, USA, ACM, 2004, pp. 55–64, http://dx.doi.org/10.1145/1029208.1029217, URL http://doi.acm.org/10.1145/1029208.1029217.

[12] J.R. Goodall, W.G. Lutters, P. Rheingans, A. Komlodi, Preserving the big picture: Visual network traffic analysis with TN, in: K. Ma, S.C. North, W. Yurcik (Eds.), IEEE Workshop on Visualization for Computer Security (VizSEC 2005), 26 October 2005, Minneapolis, MN, USA, IEEE Computer Society, 2005, p. 6, http://dx.doi.org/10.1109/VIZSEC.2005.17.

[13] A. Yelizarov, D. Gamayunov, Visualization of complex attacks and state of attacked network, in: D.A. Frincke, C. Gates, J.R. Goodall, R.F. Erbacher (Eds.), 6th International Workshop on Visualization for Cyber Security 2009, VizSec 2009, Atlantic City, New Jersey, USA, October 11, 2009, IEEE Computer Society, 2009, pp. 1–9, http://dx.doi.org/10.1109/VIZSEC.2009.5375527.

[14] M. Angelini, N. Prigent, G. Santucci, PERCIVAL: proactive and reactive attack and response assessment for cyber incidents using visual analytics, in: L. Harrison, N. Prigent, S. Engle, D.M. Best (Eds.), 2015 IEEE Symposium on Visualization for Cyber Security, VizSec 2015, Chicago, IL, USA, October 25, 2015, IEEE Computer Society, 2015, pp. 1–8, http://dx.doi.org/10.1109/VIZSEC.2015.7312764.

[15] S. McKenna, D. Staheli, C. Fulcher, M.D. Meyer, Bubblenet: A cyber security dashboard for visualizing patterns, Comput. Graph. Forum 35 (3) (2016) 281–290, http://dx.doi.org/10.1111/cgf.12904.

[16] M. Tavallaee, E. Bagheri, W. Lu, A.A. Ghorbani, A detailed analysis of the KDD CUP 99 data set, in: 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications, CISDA 2009, Ottawa, Canada, July 8-10, 2009, IEEE, 2009, pp. 1–6, http://dx.doi.org/10.1109/CISDA.2009.5356528.

[17] N. Moustafa, J. Slay, The evaluation of network anomaly detection systems: Statistical analysis of the UNSW-nb15 data set and the comparison with the KDD99 data set, Inform. Secur. J.: Global Perspect. 25 (1–3) (2016) 18–31, http://dx.doi.org/10.1080/19393555.2015.1125974.

[18] W. Zong, Y.-W. Chow, W. Susilo, A 3d approach for the visualization of network intrusion detection data, in: 2018 International Conference on Cyberworlds (CW), 2018, pp. 308–315, http://dx.doi.org/10.1109/CW.2018.00064.

[19] T. Janarthanan, S. Zargari, Feature selection in UNSW-nb15 and KDDCUP'99 datasets, in: 2017 IEEE 26th International Symposium on Industrial Electronics (ISIE), 2017, pp. 1881–1886, http://dx.doi.org/10.1109/ISIE.2017.8001537.

[20] P. Sangkatsanee, N. Wattanapongsakorn, C. Charnsripinyo, Practical real-time intrusion detection using machine learning approaches, Comput. Commun. 34 (18) (2011) 2227–2235, http://dx.doi.org/10.1016/j.comcom.2011.07.001.

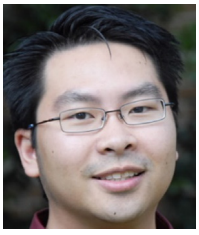
[21] E. de la Hoz Correa, E. de la Hoz Franco, A. Ortiz, J. Ortega, B. Prieto, PCA filtering and probabilistic SOM for network intrusion detection, Neurocomputing 164 (2015) 71–81, http://dx.doi.org/10.1016/j.neucom.2014.09.083.

[22] A. Lakhina, M. Crovella, C. Diot, Diagnosing network-wide traffic anomalies, in: R. Yavatkar, E.W. Zegura, J. Rexford (Eds.), Proceedings of the ACM SIGCOMM 2004 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication, August 30 - September 3, 2004, Portland, Oregon, USA, ACM, 2004, pp. 219–230, http://dx.doi.org/10.1145/1015467.1015492, URL http://doi.acm.org/10.1145/1015467.1015492.

[23] J. Camacho, A. Pérez-Villegas, P. García-Teodoro, G. Maciá-Fernández, PCA-Based multivariate statistical network monitoring for anomaly detection, Comput. Secur. 59 (2016) 118–137, http://dx.doi.org/10.1016/j.cose.2016.02.008.

[24] P.A.A. Resende, A.C. Drummond, A survey of random forest based methods for intrusion detection systems, ACM Comput. Surv. 51 (3) (2018) 48:1–48:36, http://dx.doi.org/10.1145/3178582.
[25] P.E. Rauber, S.G. Fadel, A.X. Falcão, A.C. Telea, Visualizing the hidden activity of artificial neural networks, IEEE Trans. Vis. Comput. Graph. 23 (1) (2017) 101–110, http://dx.doi.org/10.1109/TVCG.2016.2598838.
[26] L. Van Der Maaten, Accelerating t-SNE using tree-based algorithms, J. Mach. Learn. Res. 15 (1) (2014) 3221–3245.
[27] Z. Ruan, Y. Miao, L. Pan, N. Patterson, J. Zhang, Visualization of big data security: a case study on the kdd99 cup data set, Digit. Commun. Netw. 3 (4) (2017) 250–259, http://dx.doi.org/10.1016/j.dcan.2017.07.004, URL http://www.sciencedirect.com/science/article/pii/S2352864817300810, Big Data Security and Privacy.
[28] I. Onut, A.A. Ghorbani, Svision: A novel visual network-anomaly identification technique, Comput. Secur. 26 (3) (2007) 201–212, http://dx.doi.org/10.1016/j.cose.2006.10.001.
[29] E. Corchado, Á. Herrero, Neural visualization of network traffic data for intrusion detection, Appl. Soft Comput. 11 (2) (2011) 2042–2056, http://dx.doi.org/10.1016/j.asoc.2010.07.002.
[30] J. McHugh, Testing intrusion detection systems: a critique of the 1998 and 1999 DARPA intrusion detection system evaluations as performed by lincoln laboratory, ACM Trans. Inf. Syst. Secur. 3 (4) (2000) 262–294, http://dx.doi.org/10.1145/382912.382923, URL http://doi.acm.org/10.1145/382912.382923.
[31] M. Liu, J. Shi, K. Cao, J. Zhu, S. Liu, Analyzing the training processes of deep generative models, IEEE Trans. Vis. Comput. Graph. 24 (1) (2018) 77–87.
[32] L. Wei, Multi-class blue noise sampling, ACM Trans. Graph. 29 (4) (2010) 79:1–79:8, http://dx.doi.org/10.1145/1833351.1778816.
[33] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: Machine learning in python, J. Mach. Learn. Res. 12 (2011) 2825–2830, URL http://dl.acm.org/citation.cfm?id=1953048.2078195.
[34] G. Kim, S. Lee, S. Kim, A novel hybrid intrusion detection method integrating anomaly detection with misuse detection, Expert Syst. Appl. 41 (4) (2014) 1690–1700, http://dx.doi.org/10.1016/j.eswa.2013.08.066.

Wei Zong obtained his BCompSc. from the University of Wollongong, Australia. He is currently an MPhil candidate in the University of Wollongong. His research interests include machine learning, intrusion detection and visualization.



Yang-Wai Chow received his BSc., B.Eng. (Hons.) and Ph.D. from Monash University, Australia. He is currently an associate professor in the School of Computing and Information Technology, at the University of Wollongong, Australia. His research interests include virtual reality, interactive real-time interfaces, multimedia security and cyber security.

Willy Susilo obtained his Bachelor Degree in Computer Science from Universitas Surabaya, Indonesia with a "Summa Cum Laude" predicate. He received his Master and Doctor of Philosophy degrees from the University of Wollongong (UOW). His main research interest include cryptography and cyber security. He received a prestigious ARC Future Fellowship from the Australian Research Council. He also received the UOW Researcher of the Year 2016 due to his research excellence. He is the Director of Institute of Cybersecurity and Cryptology, UOW.