2022

A Validity-Based Approach for Feature Selection in Intrusion Detection Systems

Eljilani Hmouda
Nova Southeastern University, eh755@nova.edu

A Validity-Based Approach for Feature Selection in Intrusion Detection Systems

by

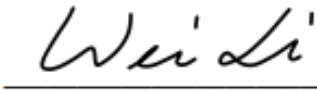
Eljilani Hmouda

A dissertation submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy
in
Computer Science

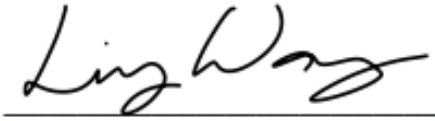
College of Computing and Engineering
Nova Southeastern University

2022

We hereby certify that this dissertation, submitted by Eljilani Hmouda conforms to acceptable standards and is fully adequate in scope and quality to fulfill the dissertation requirements for the degree of Doctor of Philosophy.



Wei Li, Ph.D. — **5/17/22**
Chairperson of Dissertation Committee — **Date**



Ling Wang, Ph.D. — **5/17/22**
Dissertation Committee Member — **Date**

Ajoy Kumar, Ph.D. — **5/17/22**
Dissertation Committee Member — **Date**

Approved:

Meline Kevorkian, Ed.D. — **5/17/22**
Dean, College of Computing and Engineering — **Date**

An Abstract of a Dissertation Submitted to Nova Southeastern University
in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy

A Validity-Based Approach for Feature Selection in Intrusion Detection Systems

by
Eljilani Hmouda
May 2022

Intrusion detection systems are tools that detect and remedy the presence of malicious activities. Intrusion detection systems face many challenges in terms of accurate analysis and evaluation. One such challenge is the involvement of many features during analysis, which leads to high data volume and ultimately excessive computational overhead. This research surrounds the development of a new intrusion detection system by employing an entropy-based measure called v-measure to select significant features and reduce dimensionality. After the development of the intrusion detection system, this feature reduction technique was tested on public datasets by applying machine learning classifiers such as Decision Tree, Random Forest, and AdaBoost algorithms. We have compared the results of the features selected with other feature selection techniques for correct classification of attacks. The findings demonstrated dimension and data volume reduction while maintaining low false positive rate, low false negative rate, and high detection rate.

Acknowledgments

The long and arduous journey of completing my PhD could not have been completed without the support of my family. I am deeply grateful for my father (may he rest in peace) who had provided me with a huge amount of encouragement to complete my degree. My utmost gratitude goes to my mother and my wife Biya, my kids Mohammed, Abdulhamid, Rahaf, and Nour, for their patience and support during this journey. They surrounded me with love and tenderness all the time. Despite all difficulties I faced, they never gave up on me. I would also like to express my gratitude and appreciation to my brothers Abdulmajid and Mansour for their love and support.

Utmost gratitude is owed to my adviser Dr. Wei Li for his patience and his support. His valuable comments were the light that lit my way. I would like to thank my dissertation committee Dr. Ling Wang and Dr. Ajoy Kumar for their advice and guidance.

In addition, I would like to thank my friends Osama Faker, Milad Abujarada, and Mohammed Almutaz, for their advice and consultation. Last but not least, I would like to thank all my friends and NSU colleagues who supported me to complete my degree.

Table of Contents

List of Tables

Tables

List of Figures

Figures

Chapter 1

Introduction

Background

An intrusion is outlined as any sequence of actions that compromise the confidentiality, integrity, or availability of a network or a host. Intrusion detection is a process of tracking and monitoring the passing of information and identifying malicious activities. Intrusion detection systems (IDS) are defense systems that detect anomalous activities. A network IDS has the capability to provide an overview of unusual behavior and issue alerts to inform the network administrators and terminate a suspected connection (Zhang & Lee, 2000). Intrusion detection systems function through either network-based or host-based intrusion detection techniques (Ahmed et al., 2016).

Network intrusion detection systems (NIDS) track the incoming data from several resources and detect malicious activities targeted on networking resources, while Host-based intrusion detection system (HIDS) is capable of tracking and analyzing data on computers (Kozushko, 2003). An IDS can also be categorized according to its detection methods. A signature-based system detects attacks based on comparing network data to known attacks, which are labeled as signatures in a database. An anomaly-based system detects intrusions based on deviations from normal user activities and is able to detect novel attacks without prior knowledge (Patcha & Park, 2007). In a network-based IDS, in order to identify anomalous traffic to networks, all packets throughout the network are classified to verify if the data contains any malicious activity or not.

Traffic information is represented in records and attributes (sometimes referred to as features). The huge volume of data serves as one of the key challenges for anomaly detection (Chandola et al., 2009).

Therefore, the analysis of this information obligates the use of statistical methods and data mining techniques for feature selection. The objective of feature selection is to avoid selecting irrelevant features and use only the most significant features that represent the packet activity (Lima et al., 2012).

Feature selection is a process of selecting relevant attributes that assist in understanding data, reducing the computational requirements, and improving the performance of the predictor. The main focus of feature selection is to select a set of relevant variables from input data while reducing the effects of irrelevant variables and obtain good prediction results. By removing insignificant variables, the number of features can be reduced which can lead to the reduction of overfitting, the reduction of training time, and the improvement of accuracy. For this purpose, the feature selection criterion is vital to measure each feature according to the output class and labels (Chandrashekar & Sahin, 2014). Many feature selection methods have been discussed in the literature based on machine learning algorithms such as genetic algorithm (Haury et al., 2011), sequential selection algorithms (Reunanen, 2003), clustering algorithms (Law et al., 2004), and Ensemble feature selection (Haury et al., 2011). There are no efficient solutions to date to select the best features and detect every type of anomalous activity as the feature selection method is highly specific to attacks and datasets.

Feature selection is not only utilized to reduce the dimensionality of the data, it also commonly contributes to a more compact model with higher generalization capability (Solorio-Fernández rt al., 2019).

In this dissertation research, we adopted an entropy-based measurement termed v-measure. Entropy is defined as a measure on the homogeneity and completeness of a dataset (Rosenberg & Hirschberg, 2007). A clustering result satisfies homogeneity when each cluster must assign only datapoints that are members of a single class. A clustering result satisfies completeness when all datapoints that are members of a given class are members to the same cluster. It informs us how much additional information we would get from understanding the feature's value (Marsland, 2014).

Feature selection can be viewed as a clustering problem when the significance of individual features or subset of features is evaluated based on the clusters they produce. For example, for a binary anomaly detection problem, when a feature is used to cluster a training dataset into two clusters – benign and attack, and the clusters are a prefect partition of intrusions against non-intrusions, the feature is considered a significant feature in anomaly detection (Liu & Yu, 2005; Roth & Lange, 2003).

V-measure has been a frequently used method to measure the quality of clustering results. For example, a Normalized Mutual Information (NMI) technique that equivalents to v-measure metric has been used to effectively identify trending events in social media (Becker, 2011). In another application, v-measure has been implemented as an evaluation method to assess the clustering quality for a statistical method called PyClone, for the conjecture of clonal population structures in cancers (Roth et al., 2014). According to Yin and Wang (2014), a collapsed Gibbs Sampling algorithm for short text clustering was proposed, in which v-measure has been deployed to evaluate the homogeneity and completeness of the clustering quality.

In the research by Rosenberg and Hirschberg (2007), it has been shown that v-measure overcomes the clustering issues such as dependency and the problem of matching. V-measure

doesn’t depend on the number of classes, the number of clusters, and the clustering algorithm. A common problem in clustering is that only a portion of cluster membership is evaluated because of the problem of matching (Rosenberg & Hirschberg, 2007). However, v-measure provides an elegant solution by measuring the relative sizes of the clusters and classes being evaluated; thus, v-measure evaluates the entire membership of every single cluster and not just a matched portion. In addition, v-measure provides an objective evaluation on features by combining two clustering aspects which are homogeneity and completeness. Homogeneity is maximized when all the clusters contain data points that only belong to a single class. Completeness is maximized when all the data points of a single class belong to a single cluster (Rosenberg & Hirschberg, 2007).

Two experiments have been conducted for evaluation, document clustering experiment, and a pitch accent type clustering experiment by using three entropy-based measurements: v-measure, Q0 (Dom, 2002), and variation of information (VI) (Meila, 2007). In these experiments, it has been shown that v-measure excels Q0 in that it evaluates the completeness of features, while Q0 does not. On the other hand, v-measure excels VI in that VI requires dependency on the number of datapoints being clustered which v-measure does not. Another advantage of v-measure over Q0 and VI is that the calculation of v-measure makes the relationship between homogeneity and completeness discernible.

By this research, we studied the impact of feature selection to detect network anomalies, to reduce false positive and false negative rates, and to improve the accuracy of network intrusion detection. Through our experiments, we believed that v-measure is a good candidate for all these research goals and is worthy of further exploration.

Problem Statement

Intrusion detection system normally deals with a huge amount of redundant and irrelevant features, which may cause high positive rate and low detection rate. Removing irrelevant features results in a better performing model, reduces computation time, and presents an easy-to-understand model. In this research, intrusion detection problem is simplified as a binary classification problem, which includes one class as normal (benign) and another class as abnormal (attack). The normal class is assigned class label (0) and the class with abnormal is assigned the class label (1). This simplification suggests that the main focus of this research is not simply on anomaly detection, but rather on the effectiveness of v-measure as a good candidate in for feature selection. Due to this simplification, it is likely that less effective features may be chosen and do not fit well for multi-label classifications (i.e., classifying attacks in multiple categories). However, as a generic technique, we believe that v-measure can be extended to multi-label classification tasks in anomaly detection after fine-tuning.

Entropy is an external measure that is commonly used for clustering evaluation. The entropy of a cluster shows from one perspective how the datapoints are distributed within each cluster. It has been shown by the literature that v-measure outperforms entropy by computing both homogeneity and completeness of a giving cluster in the field of natural language processing tasks, such as document clustering (Zhao & Karypis, 2001).

Despite the significance of feature selection in anomaly detection, there was no perfect solution in feature selection. In this research we have used v-measure as a feature selection technique for network-based anomaly detection. The research problem can be formally represented as follows. Given a set of features $F = \{f_1, f_2, \dots, f_i \dots, f_n\}$, where n is the total number of features in the original dataset. By evaluating the significance of v-measures of each

feature, a subset of $F' = \{f_{v_1}, f_{v_2}, \dots f_{v_j}, \dots, f_{v_l}\}$, can be identified and verified for anomaly detection, where $v_j \in F$ $(1 \leq j \leq l)$ is a selected significant feature, and l is the number of selected features.

The chosen subset of feature was tested by machine learning algorithms such as Decision Tree, Random Forest, and AdaBoost based on benchmark datasets of wired network traffic. The results have been presented using confusion matrices, which include true positives, true negatives, false positives, and false negatives.

This research methodology has been evaluated using a recent intrusion detection system dataset CICIDS2017 (Sharafaldin et al., 2018) after three main steps. Firstly, modifying the dataset and removing missing values to ensure that the dataset is free of incorrect and irrelevant information. Secondly, applying v-measure among all features and select the features that have better v-measure scores. Finally, deploying three classifiers to evaluate the quality of chosen features in terms of data reduction rate, detection rate, and false positive rate.

Dissertation Goal

In this research, the focus is to improve the performance of anomaly-based intrusion detection system and reduce feature dimensionality using v-measure. The main goals of this research are as follows:

1. Adopt v-measure to select significant features and reduce data dimensionality for anomaly detection.
2. Maintain low false positive and false negative rate and high detection rate in binary intrusion classification.

3. Compare classification results using the features produced by v-measure with other feature selection techniques. The results have been evaluated and presented by confusion matrix (includes true/false positives and true/false negatives).

Research Questions

1. Is v-measure a good feature selection technique in improving intrusion detection based on the CIDISD2017 dataset, while maintaining high detection rate and low false positive and false negative rate at the same time?
2. What are the computational costs of v-measure when it is compared to other statistical measures such as F-measure or Information Gain?

Relevance and Significance

There is no commonly agreed approach to improve the intrusion detection system to reduce data dimensionality and false positives. In many situations, feature selection has the potential to reduce redundant data and improve detection accuracy, as shown in research by Watson (2018), based on CICIDS2017 dataset.

The proposed dissertation research concentrates on some techniques that extract features from a large dataset with the goal of dimensionality reduction. It is assumed that the advantages of reducing the features are as follows (Chandrashekar & Sahin, 2014):

- Improve the machine learning algorithm's parameters.
- Reduce computational time and storage space.
- Simplify data presentation by focusing on a subset of key features.

Validity measure (v-measure) has been introduced by Rosenberg and Hirschberg (2007). V-Measure was developed to tackle the issue of evaluating the clustering results. In other words, it is one way to evaluate the clustering results that could provide the suitable features to enhance

the intrusion detection system. As shown in the literature review, v-measure has been widely used as a clustering evaluation metric in diverse applications. We have used v-measure in the classification and selection process, and evaluated which features provide better false alarm reduction and better accuracy. V-measure has helped to identify appropriate features of the incoming traffic, whose tracking would ensure reliable detection of anomalous activity. It also determines the connection of those features to the intrusion detection task and the features' redundancy. V-measure is therefore a good candidate for classifying network traffic with high accuracy result and low false positive and false negative rate.

Based on our experiments, v-measure not only identifies important features in the tested dataset, but also help reducing false positive and false negative rates and improving detection rate. We believe this is a contribution to the body of knowledge in anomaly-based IDS.

Barriers and Issues

One of the main challenges for this research is to extract a set of features to represent the activity of the collected data based on a specific measure. Feature selection has been shown as a dimension reduction technique in dealing with high dimensional data. The main concentration of the feature selection is to select a suitable subset of variables from huge dataset. This process has been done by eliminating the dependent and irrelevant variables, which can lead to enhancement in the classification performance (Li & Liu, 2017). In this research, v-measure has been implemented as a feature selection that has acted as a promising choice. There are also multiple other options such as F-measure (Van Rijsbergen, 1979), including the entropy-based measures such as Q0 (Dom, 2002), and variation of information (VI) (Meila, 2007). V-measure has been compared against other statistical measures such as F-measure and filter method such as

Information Gain (IG) on their effectiveness and efficiency of feature selection. The results have been validated using multiple machine learning classifiers.

In this research, the modeling of feature selection as a clustering problem, the ranking of individual classifiers, and the use of filter methods in general, may not be perfect ways for feature selection. Other techniques, such as the use of wrapper or embedded methods (Srihari & Anitha, 2014; Wang et al., 2015), the use of machine learning techniques directly on feature selection, may produce better results in feature selection. The results may also vary depending on the datasets adopted for testing. Despite these possibilities, we believed the proposed research is a novel path in feature selection for anomaly detection and is worthy of in-depth exploration.

Datasets are a significant portion of evaluating different IDSs. There have been many critiques on the popular KDD 99 dataset in that it lacks traffic diversity and volume, attack types, or feature sets (McHugh, 2000). Research has been made to improve the KDD99 dataset, which leads to NSL-KDD (Tavallaee et al., 2009). However, finding up to date dataset remains difficult. Moustafa and Slay (2015) have concluded that there was an absence of a suitable comprehensive data set for the evaluation of NIDS research efforts. They have created a new data set called UNSW-NB15 due to the fact that the dataset KDD99 does not include modern network traffic. In this research, CICIDS2017 was adopted as the intrusion detection system dataset.

The CICIDS2017 has many of the desirable properties that fits our research. For example, the size of the dataset (the simulation covers 5 days of attack simulation with each of a size about 10 GB), the wide coverage of a variety of network protocols, the number of attacks (e.g., Brute Force FTP, Brute Force SSH, DoS, Heartbleed, Web Attack, Infiltration, Botnet and DDoS), and most importantly, the number of features (about 80 network flow features) are some

of the desirable properties (Sharafaldin et al., 2018). We understand that it may not be the most up-to-date dataset and there could be better choices in literature, however we believe it is suitable to the goal of this research.

Detecting anomalous activity can be viewed as a binary classification problem in which measurements of system activity such as audit records are used to provide a classification of the state of the system as normal or abnormal. However, detecting anomalous activity based on multi label classification is outside the scope of this research. V-measure may be able to perform well on this specific binary classification problem, but the chance of it performing well on other more complicated detection problems may need further exploration beyond this research. The focus of this research is to test out the effectiveness of the v-measure in feature reduction while maintaining low positive rate and high detection rate in binary intrusion classification. Despite potential issues, one key benefit of using binary classification is the reduced computational cost. This is a significant issue because during the feature selection process, we need to apply v-measure on each of the features in a large dataset.

Assumptions, Limitations, and Delimitations

Intrusion detection systems data (CICIDS-2017) was collected from diverse sources which fetches numerous data types and complex data structure. There are two assumptions presented in this research. The first assumption is the quality of the dataset where each feature provides accurate information. The second assumption is about the consistency in the significance and relationship of the data across the features. This has been noted during the implementation of the proposed method to possibly make future expansions simpler.

The main limitation of this research is the volume of the data. As a consequence, the high dimensional dataset leads to long classification processes. Another expected limitation incurred computational cost, as noted in the previous section.

The delimitations of this research are concentrated on developing a new feature selection to detect network anomalies, reduce false positive and false negative rates, and improve the accuracy of network intrusion detection. It should be note that there is no perfect methodology in feature selection for anomaly detection despite the outcomes of this research. Any methodology is limited by the statistical nature of the feature selection techniques, the number and nature of features, the datasets being tested, the nature of the attacks, the computational cost, and could vary case by case - sometimes significantly.

Definition of Terms

Intrusion: any sequence of actions that makes an attempt to compromise the confidentiality, integrity, or availability of the information in a network or a host.

Intrusion detection systems (IDS): tools that detect and remedy the presence of malicious activities.

Network-based intrusion detection systems (NIDS): a system that tracks the incoming data from several resources and detect malicious activities targeted at networking resources.

Host-based intrusion detection system (HIDS): a system that tracks is capable of tracking and analyzing data on computers.

False Positive: the identification of an activity as an attack while the activity is normal behavior.

False Negative: the identification of an activity as normal behavior while the activity is an attack.

Feature Selection: a process of selecting relevant attributes that assist in understanding data, reducing the computational requirements, and improving the performance of the predictor.

Entropy: an external measure that evaluates the impurity of data.

Validity Measure (V-measure): an external measure that utilized as a frequently method to measure the quality of clustering results based on homogeneity and completeness concepts.

Variation of Information (VI): a linear expression that involves the mutual information by measuring the distance between two clusters.

List of Acronyms

IDS: Intrusion Detection Systems

NIDS: Network-based intrusion detection systems

HIDS: Host-based intrusion detection system

V-measure: Validity Measure

NMI: Normalized Mutual Information

PyClone: Bayesian clustering method

VI: Variation of Information

Q0: Dom's Q0 measure

SVM: Support Vector Machine

DT: Decision Tree

RF: Random Forest

AC: Accuracy

FP: False Positive

FN: False Negative

TP: True Positive

TN: True Negative

DR: Detection Rate

Summary

This chapter provided an overview of the issues that face anomaly-based intrusion detection systems in terms of precise analysis and evaluation. This research is about deploying new entropy metric as a feature selection technique. Analyzing a set of data collected from intrusion detection systems is a difficult task due to the huge number of features. Thus, it's very important to present an efficient method to reduce the number of features for intrusion detection while maintaining low false positive rate, low false negative rate, and high detection rate. In this research, we have used v-measure as a feature selection technique and have tested its performance against other similar methods in the literature.

Chapter 2

Review of the Literature

Intrusion Detection Systems

The idea behind an intrusion detection system can be traced back to Anderson (1980), who proposed a set of tools for threat monitoring and surveillance. Later Denning (1987) developed the first intrusion detection model. The model was based on a system's audit records that can be controlled by security violations in order to obtain normal patterns of system application.

Anomaly Detection

Network-based intrusion detection attempts to track the network traffic from multiple resources, detect malicious behavior affecting multiple hosts, and protect the entire network (Kozushko, 2003). An anomaly is defined as a pattern that doesn’t adapt to normal behavior. Despite that anomaly detection has the potential to detect novel attacks, a significant problem is the huge number of alerts and some of them are false positive (Zuech et al., 2015). As indicated by the literature, one major challenge in anomaly detection research is the huge data volume for accurate analysis, and the other is the reduction of false positive rates (Chandola et al., 2009). Due to the complexity of intrusion detection, these alerts sometimes have to be analyzed and evaluated by a human analyst to reduce their volume. As a consequence, identifying the false positive is an important task for analysts due to the abundance of the alerts in the alert log (Pietraszek, 2004).

In this section, we will briefly mention the contemporary key references that illustrate the research motivation and emphasize the limitations in the prior literature (2004-2020).

A number of anomaly-based IDS have been built on different types of machine learning algorithms such as genetic algorithm (Li, 2004), neural network (Mishra et al., 2018), decision tree (Lee et al., 2008), support vector machine (Sotiris et al., 2010), and ensemble methods (Vanerio & Casas, 2017), where most of them are assessed and evaluated with KDDCup99 and NSL-KDD (Tavallaee et al., 2009).

Selecting suitable features is a significant issue in intrusion detection (Li et al., 2017). Several approaches have tried to address the problem of selecting suitable features, and reduce the data volume (Das, 2001). For instance, many machine learning algorithms and statistical methods including entropy are utilized in a variety of analysis techniques for recognizing attacks, (Zhao & Karypis, 2001).

The research conducted by Stein et al. (2005) introduced a feature selection approach based on a genetic algorithm to select subsets as an input for decision tree classifier. The hybrid approach concentrated on using relevant features to increase the detection rate and to reduce false positive alerts. However, the execution time was longer than using the standard decision tree algorithm. Moreover, the support vector machine was utilized as a feature selection technique to improve the naïve Bayes classification (Thomas & Peter, 2001).

Chiu et al. (2010) proposed another filter technique by using semi-supervised learning technique called Two-Teachers-One-Student algorithm (2T1S) to obtain important information from unlabeled data. The experiments were conducted using DARPA intrusion detection evaluation dataset. DARPA was generated by Defense Advanced Research Projects Agency and Air Force Research Laboratory in 1998, and was managed by MIT Lincoln Labs. They adopted the DARPA 1999 dataset for experiments because of its diversity of attacks. It was extracted using a network connection feature extractor based on TCP connection. In the first stage, a

statistical connection feature method was implemented to reduce the number of false alarms. Semi-supervised learning algorithm was applied on unlabeled data to improve the performance. The combined approach was capable to filter out 65% false alarms with missing less than 0.1% of true attacks in the filtering alarms.

Machine Learning for Anomaly Detection

Some of the machine learning algorithms which are commonly used by intrusion detection systems are not used in our research because we want to focus specifically on the use of statistical methods for feature reduction (Liu & Lang, 2019). One of the popular approaches in anomaly detection is the use of neural networks. Neural networks have the capability to analyze incomplete data from the network and to predict detection for any attack (Mishra et al., 2018). Another acclaimed machine learning technique used in intrusion detection systems is a genetic algorithm. It is used for finding optimal solutions space and doesn't need prior knowledge of the problem. In some approaches, genetic algorithms were implemented to determine classification rules for misuse detection (Li, 2004), while other approaches have implemented genetic algorithms as a feature selection to select suitable features or determine optimal parameters of relevant functions (Hira & Gillies, 2015). In terms of unsupervised intrusion detection, data clustering methods can be used. These methods include calculating a distance between numeric features; therefore, they cannot deal with symbolic attributes, ending in inaccuracy (Jha & Ragha, 2013).

In addition, data mining techniques were used to investigate the root cause behind large numbers of false alarms, and machine learning was used to classify and distinguish true and false positive rates (Pietraszek & Tanner, 2005). It was stated that the number of alerts could be reduced by removing the most important root causes. The problem with this approach was the

dependency on the human interpretation, which has limited its practical use. A previous study by Parikh and Chen (2008) presented a hybrid approach by combining ensemble classifiers to derive information from different sources. A cost minimization strategy was applied to minimize the true object function and the classification error cost. It was stated that the approach was capable to reduce the false positive alarm and provide better enhancement.

Malhi and Gao (2004) claimed that feature reduction has the potential to help increase the detection rate and reduce the false alarm rate. Principal Component Analysis (PCA) has been implemented for both supervised and unsupervised classification. They present the scope of their research by attempting an implementation of feedforward neural network and radial basis function network, and unsupervised competitive learning. They confirmed the experiment's results by using cross validation, wherein the selected features by PCA increased the accuracy and detection rate.

In the following discussion, we will review three specific supervised learning classifiers because they have been used for the evaluation of feature selection tasks in our research. The first one, Decision Tree (DT), was implemented due to its efficiency dealing with large datasets without implementing a complicated parametric structure. Decision Tree is a graphical representation consisting of internal nodes and branches that presents all potential solutions to a decision based on particular conditions (Song & Ying, 2015). The tree is constructed by identifying features and their associated values which has been utilized to analyze the entered data at each intermediate node of the tree. Decision trees have the capability to analyze data and identify important factors in the network that indicate abnormal activities (Rai et al., 2016).

The main advantages of utilizing decision tree in this research is its simplicity to handle large data without requiring data transformation, splitting original input variables into subgroups

makes the relationships between input variables and target variables simpler (Song & Ying, 2015).

The second classifier that was used in our research, the random forest, is a classification algorithm that generates multiple trees and merges them to gain an accurate prediction (Mishra et al., 2018). After the forest is structured, each new node needs to be classified under the three types of nodes root node, decision node, or leaf node. Each tree votes to make a decision about the class it joined. The majority votes for the objects were selected by the forest. The Random Forest model is an ensemble of Decision Trees, which could be utilized for regression or classification. In regression, the average of the tree's output is considered as a result; while in classification, decision tree determines the votes of the predicted value (Resende, & Drummond, 2018). We choose Random Forest as a classifier to evaluate our new feature selection based on its strengths when compared to the different machine learning algorithms. The key advantages of Random Forest are: (i) low training time complexity $O(nlog(n))$ and rapid prediction; (ii) flexibility to deal with unbalanced datasets; and (iii) ranking features by importance (Witten et al., 2016; Khoshgoftaar et al., 2007).

The above-mentioned advantages revealed the efficiency of Random Forest when involved in a comparative study in the domain of intrusion detection systems. For instance, ten classification algorithms namely J48, BayesNet, Logistic, SGD, IBK, JRip, PART, Random Forest, Random Tree and REPTree have been deployed using NSL-KDD dataset based on four classification metrics. The experiments showed that Random Forest achieved better performance (Chauhan et al., 2013). The following studies have indicated significant performance for Random Forest models (Aziz et al., 2017; Amudha, & Rauf, 2011; Robinson, & Thomas, 2015).

The third classifier, the Adaptive Boost (AdaBoost), is a supervised machine learning model that is well-known for its performance in pattern recognition and binary classification problems. AdaBoost has provided better performance than traditional learning machines and has been widely applied in real-world classification problems and nonlinear function estimation problems (Hu et al., 2008). In our research, AdaBoost has been implemented due to its speed for detecting intrusion activities and low computational complexity. AdaBoost has been applied to assurance security for intrusion detection due to its real generalization nature and the capability to defeat the imprecation of dimensionality. Another positive aspect of AdaBoost is the capability to update the training patterns during the classification process (Chu et al., 2020).

Deep Learning for Anomaly Detection

Deep learning is a branch of machine learning, which offers numerous benefits when it applies to anomaly detection. These approaches are developed to work with high-dimensional data and obtain outstanding performances. This makes it easy to combine data from various sources, and reduces challenges linked with independently modeling anomalies for every variable and aggregating the outcomes (Liu & Lang, 2019).

Deep learning models consist of several deep networks that are supervised learning models such as Deep Brief Networks (DBNs), Deep Neural Networks (DNNs), Convolutional Neural Networks (CNNs), and Recurrent Neural Networks (RNNs). While some unsupervised learning models such as autoencoders, Restricted Boltzmann Machines (RBMs), and Generative Adversarial Networks (GANs). Deep learning approaches are adaptable to modeling the interactions between many variables according to a given task, deep learning requires minimum tuning to obtain good results aside from the number of layers and units per layer (Liu & Lang, 2019). Deep learning approaches provide modeling of nonlinear relationships that can be utilized

in detecting anomaly behavior, and have been used in a number of approaches (Pang et al., 2021).

Feature Selection Techniques

There are a number of feature selection techniques that have been proposed in the literature. Feature selection methods are generally classified into three categories: filter, wrapper, and embedded methods.

Filter methods select features by using the variable ranking techniques such as the principal criteria (Das, 2001). Filter ranking methods mentioned in the literature include Fisher Score where features are selected independently based on fisher criterion (Duda et al., 2012), Pearson Correlation which computes linear correlation between two variables (Miyahara & Pazzani, 2000), and Mutual Information (MI) which measures the quantity of information that one random variable includes about another random variable (Battiti, 1994). These methods do not consider the dependency among features, and multiple correlated features have been selected by the feature selection algorithm.

Alharby and Imai (2005) proposed a filtering technique to reduce false alarms rates. The alarms' sequential patterns were classified into two classes, continuous and discontinuous. The similarity algorithm was calculated to find the similarity between sequential patterns and normal sequence patterns. When the system is under attack, it acts in a different way from that in the normal situation. The system can distinguish between alarms when the system is under attack or in a normal situation. The network traffic was examined using Snort (Roesch, 1999).

Wrapper methods use machine learning algorithms to find the most suitable features based on three methods such as forward selection, backward selection, and stepwise selection

(Das, 2001). These methods are sometimes computationally expensive than filter methods due to the use of machine learning algorithms.

Embedded methods are a combination of the filter methods and wrapper methods where feature selection process and classification are performed simultaneously (Chandrashekar & Sahin, 2014). In embedded methods, an independent measure based on wavelet analysis has been used to select suitable features and machine learning algorithms to decide the final features (Srihari & Anitha, 2014; Wang et al., 2015).

Entropy-Based Methods

Entropy-based methods were used to select important features in IDS (Wang et al., 2015). The filter model was designed based on C4.5 decision tree. Three entropy methods such as Shannon entropy, Renyi entropy, and Tsallis have been applied to select the features on KDD99 dataset. It was stated that the selected features based on Renyi and Tsallis entropies provided better results but increased the computational time. Another effective feature selection approach based on Bayesian network algorithm was proposed to identify the subset features using a sequential search strategy on NSL-KDD dataset (Zhang & Wang, 2013). The extracted features have been compared with three methods such as Information Gain, ChiSquare and ReliefF methods. It was claimed that the approach consumed less time than Information Gain, ChiSquare and ReliefF to detect attack and increased the classification accuracy.

Previous studies have used classifiers to evaluate feature selection techniques. The research conducted by Sindhu et al. (2012) aimed on the detection of anomalous activities. The proposed approach was based on wrapper-based feature selection that includes three classifiers such as genetic algorithm, neural network, decision tree, and random forest to identify the important features. Genetic algorithm was deployed to extract features. Neural network was used

to analyze the data to find the relationship between false positive and false negative error, and to find the suitable weight value to decrease the total error. Decision tree was utilized to classify the network traffic. The combination of the three classifiers is capable to detect certain attacks such as Neptune, Back, Smurf, and Buffer_overflow.

Another clustering evaluation measure commonly used is F-measure (Hand & Christen, 2018) which evaluates the accuracy by computing the weighted harmonic mean of precision and recall. Precision is the ratio between relevant instances and the retrieved instances, recall is the ratio between the relevant instances among actual instances. Some of the significant issues are that F-measure provides the same importance in terms of precision and recall, and neglects the true negative, which leads to misclassification.

V-measure

Rosenberg and Hirschberg (2007) developed an external entropy-based cluster evaluation technique called v-measure. V-measure addressed some of the issues that concern cluster measures: 1) V-measure evaluates the independence of the size of the dataset, the clustering algorithm, the number of clusters, and the number of classes. 2) In terms of evaluation, it does not require a mapping from clusters to classes; therefore, it only evaluates the quality of clustering. 3) It avoids the problem of matching, by evaluating the clustering of every datapoint. V-measure provides an accurate evaluation by combining two clustering aspects which are homogeneity and completeness. A clustering result satisfies homogeneity when all the clusters contain only data points that are members of a single class, where a clustering result satisfies completeness if all the data points that are members of a given class are members to a single cluster. V-measure is computed as the harmonic mean which is complementary of the arithmetic mean of homogeneity and completeness scores (Rosenberg & Hirschberg, 2007). V-measure is

more comprehensive than Q0 (Dom, 2002), and variation of information (VI) (Meila, 2007) that evaluate only homogeneity or completeness separately. In addition to the straightforward way to use v-measure to identify the significance of individual features, there are many other ways to extend its usage, for instance, adopting a pre-defined set of features, recombining existing features, and testing the significance of a group of features together.

V-measure has been widely used as a clustering evaluation metric in different applications. For example, a Normalized Mutual Information (NMI) technique that equivalents to v-measure metric has been used to effectively identify trending events in social media (Becker, 2011). In addition, another application of v-measure has been implemented as an evaluation method to assess the clustering quality of a Bayesian clustering method called PyClone, for the conjecture of clonal population structures in cancers (Roth, et al, 2014). Although v-measure has been shown as an effective feature selection technique, based on our knowledge, it has not yet been used in anomaly detection. Its effectiveness on anomaly detection is worthy of further investigation.

Intrusion Detection Datasets

Datasets are an important challenging issue in the evaluation of different IDSs. One of the most widely used dataset is KDD 99 (McHugh, 2000), originally generated under the 1998 DARPA Intrusion Detection Evaluation Program managed by MIT Lincoln Labs. However, there have been many critiques on the KDD 99 dataset due to the lack of traffic diversity and volume, lack of attack types, or lack of feature sets (McHugh, 2000). Additional research that introduced an improvement upon KDD99 dataset, and NSL-KDD (Tavallaee et al., 2009) was proposed. However, finding up-to-date dataset remains difficult.

In a later effort, Sharafaldin et al. (2018), who generated a new reliable dataset CISICD2017, which includes diverse kinds of attacks that meets real world criteria. A more recent dataset called CISICD2017 was introduced by the Canadian Institute for Cybersecurity (Sharafaldin et al., 2018). It covers diverse types of attacks such as DoS, DDoS, FTPbrute force, SSH, heartbleed, brute force, infiltration and botnet. Over 80 network traffic features have been extracted for all attacks using CICFlowMeter.

After the CISICD2017 dataset was proposed, it has been adopted in multiple research efforts. For example, CICIDS2017 was evaluated by applying seven machine learning algorithms. Random Forest has been used as a feature selection to select the best features to detect such attacks. A classical Multi-Layer Perceptron (MLP) algorithm and Convolutional Neural Network (CNN) payload classifying algorithm have been implemented on CICIDS2017. The experiment's results showed that MLP was capable of classifying malicious attacks with an average rate of 94.5% detection rate and 4.68% false positive rate (Watson, 2018).

Aksu et al. (2018) have implemented a Fisher Score algorithm as a feature selection technique on CICIDS2017, where the features are selected independently based on fisher criterion (Duda et al., 2012). The primary purpose of this approach was to apply Support Vector Machine (SVM), K-Nearest Neighbor (KNN), and Decision Tree (DT) algorithms to classify Distributed Denial of Service (DDoS) attacks. In addition, other research efforts were conducted using CICIDS2017 (Yao et al., 2019; Zhang et al., 2019).

As a result, we chose to use a revised CICIDS2017 dataset for the evaluation of the proposed v-measure. In the revised dataset, all types of attacks were classified as one class label. In our experiments, binary classification was performed based on the class labels of the CICIDS2017 dataset. This led to a simpler problem compared to a categorization problem with

different types of attacks; however, this is still a significant research problem given the volume of the dataset. The focus of this research is to test out the effectiveness of the v-measure in feature reduction. And the binary classification problem was used as a testbed for the proposed metric. We believed v-measure can also be tested in other datasets or other anomaly detection problems.

Summary

This chapter reviewed the current body of knowledge on feature selection in the domain of intrusion detection systems, as well as identified the strengths and weaknesses of the current studies and presented the gaps between them. The motivation behind this research was discussed. It also introduced an overview of some feature selection methods including their limitations. Based on the literature review, there is a need of an efficient feature selection technique that is capable to reduce the false positive and false negative while obtaining high accuracy. V-measure has been widely utilized as a clustering evaluation metric in different applications. In the following chapter, we will demonstrate how v-measure was implemented to select the suitable features in the CICIDS2017 dataset.

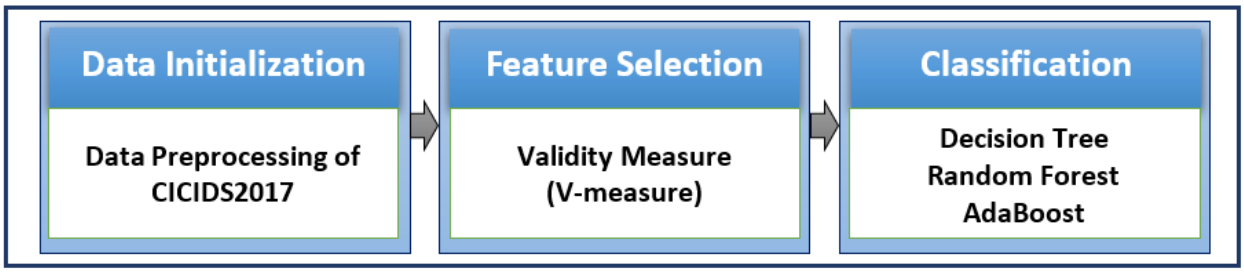
Chapter 3

Methodology

Our proposed research intends to use v-measure for feature selection in intrusion detection system. Our proposed approach was implemented in three stages. In the first stage, data has been cleaned up and transformed into an understandable format using feature scaling to standardize the independent features in a fixed scale. In the second stage, all features have been clustered using the K-means algorithm, and then v-measure has been implemented by computing the homogeneity and completeness for each cluster. The feature that has a high v-measure score was selected to build the model. In the third stage, three classifiers, Decision Tree, Random Forest, and AdaBoost were chosen to evaluate the performance of the selected features as shown in Figure 1.

Figure 1

T*he Proposed Architecture*



The proposed method introduced a novel approach that achieved high accuracy rate while reducing false positive and false negative rates. The evaluation of our proposed method used some classifiers is shown in Figure 1. In order to present the quality of our feature selection method, we have compared v-measure with other external measures such as Entropy (Zhao & Karypis, 2001). Both measures produced ways to evaluate homogeneity; however, they don't evaluate the completeness of a cluster evaluation of whether all datapoints of a class are involved

in a single cluster. Another commonly used external clustering measure is called F-measure (Van Rijsbergen, 1979). F-measure possesses a significant advantage over purity and entropy in that it does measure both homogeneity and completeness. However, the limitation of F-measure is the matching problem (Meila, 2007); therefore, v-measure has been compared against other statistical measures such as F-measure and other entropy measures such as Variation of Information (VI) on their effectiveness and efficiency of feature selection.

Data Preprocessing of CICIDS2017

In our experiment, we utilized one of the contemporary intrusion detection system datasets called CICIDS2017 (Sharafaldin et al., 2018). The data has been collected from Monday through Friday and has been exported into eight sessions. Each session has exported in the form of a comma separated value (CSV) file and labeled by its name. There are 14 types of attacks in this dataset as shown in Table 1. Over 70 network traffic features have been extracted for all attacks by using CICFlowMeter (Network Traffic Flow Generator and Analyzer Meter) (Sharafaldin et al., 2018).

The total number of records in the dataset is 2,271,320 with 79 features, of which 556,556 records are labeled as malicious traffic (Attack). While 2,273,097 records are labeled as normal traffic (Benign).

In order to perform our experiment on this data, we combined all of these 8 files into one file. We cleaned the datasets by removing records with missing values, redundant attributes, or infinite values. There are 1,358 records that have missing information and 1,509 records that have infinity value. These 2867 records that represent only 0.1 % of the 2,830,743. These records have been removed. Class label benign replaced to 0 and the class label attack replaced to 1. The total number of records after cleaning up the dataset is 2,827,876 records.

Table 1

Dataset Information

File Name	Records	Class Label
Monday-WorkingHours.pcap_ISCX	529918	Benign
Tuesday-WorkingHours.pcap_ISCX	445909	Benign, SSH-Patator, FTP-Patator
Wednesday-workingHours.pcap_ISCX	692703	Benign, DoS Hulk, DoS GoldenEye, DoS Slowloris DoS Slowhttptest, Heartbleed
Thursday-WorkingHours-Morning-WebAttacks.pcap_ISCX	170366	Benign, Web AttackBrute Force, Web Attack-Sql Injection, Web Attack-XSS 652
Thursday-WorkingHours-Afternoon-Infilteration.pcap_ISCX	288602	Benign, Infiltration
Friday-WorkingHours-Morning.pcap_ISCX	191033	Benign, Bot
Friday-WorkingHours-Afternoon-DDos.pcap_ISCX	225745	Benign, Portscan
Friday-WorkingHours-Afternoon-PortScan.pcap_ISCX	286467	Benign, DdoS
Total	**2,830,743**	**14**

Validity Measure (V-measure)

V-measure is defined as an entropy measure of homogeneity and a probability distribution of a dataset (Rosenberg and Hirschberg, 2007).

Figure 2

Calculation V-measure (Rosenberg and Hirschberg, 2007)

$$V_\beta = \frac{(1+\beta) * H * C}{(\beta * H) + C} \quad (1)$$

$H: homogeneity,\ \ C: completness,\ \ \beta: the\ ratio\ of\ precision\ and\ recall$

As defined in the equation above, the parameter β provides control over the balance between precision and recall over the score. The precision is the ratio between true positives and predicted positives. The recall is the ratio between the true positives and actual positives. This can be used to model the balance between false positives and false negatives and optimize for a more realistic score of the detection method. As a result, if β is exceeding 1, completeness is weighted more powerfully in the calculation, if β is below 1, homogeneity is weighted more powerfully. V-measure can be compared across any clustering solution, regardless of the classes' size, the clusters' size, the datapoints' size and the clustering algorithm used. Homogeneity H defined in Figure 3 and completeness C defined in Figure 4 are computed as follows.

Homogeneity

In order to satisfy our homogeneity criteria, a clustering algorithm must assign only those datapoints that are members of a single class to a single cluster. In order to calculate the cluster homogeneity, let C be the set of classes in the dataset CICIDS2017, K the set of clusters in the dataset, m the total number of elements, and a_{ck} is the number of elements from class C assigned

Figure 3

Calculation of Homogeneity in V-measure

$$H = \begin{cases} 1 & if\ H(C,K) = 0 \\ 1 - \dfrac{H(C|K)}{H(C)} & else \end{cases}$$

$where$

$$H(C|K) = -\sum_{k=1}^{|K|}\sum_{c=1}^{|C|}\frac{a_{ck}}{m}\log\frac{a_{ck}}{\sum_{c=1}^{|C|}a_{ck}}$$

$$H(C) = -\sum_{c=1}^{|C|}\frac{\sum_{k=1}^{|K|}a_{ck}}{m}\log\frac{\sum_{k=1}^{|K|}a_{ck}}{m}$$

to cluster K, as shown in Figure 3. Note that H(C|K) is maximal and equals H(C) when the clustering does not provide new information. The class distribution through each cluster is equal to the overall class distribution.

Completeness

A clustering result satisfies completeness if all the data points that are members of a given class are members to a single cluster.

Figure 4

Calculation of Completeness in V-measure

$$C = \begin{cases} 1 & if\ H(K,C) = 0 \\ 1 - \dfrac{H(K|C)}{H(K)} & else \end{cases}$$

$$where$$

$$H(K|C) = -\sum_{c=1}^{|C|}\sum_{k=1}^{|K|}\frac{a_{ck}}{m}\log\frac{a_{ck}}{\sum_{k=1}^{|K|}a_{ck}}$$

$$H(C) = -\sum_{k=1}^{|K|}\frac{\sum_{c=1}^{|C|}a_{ck}}{m}\log\frac{\sum_{c=1}^{|C|}a_{ck}}{m}$$

In order to calculate the cluster completeness, let C be the set of classes in the dataset CICIDS2017, K is the set of clusters in the dataset, m the total number of elements, and a_{ck} be the number of elements from class C assigned to cluster K, as shown in Figure 4.

V-measure Calculation Demonstration 1

From the above concepts in regard to homogeneity, completeness, and v-measure, we can turn these concepts into real examples:

Let's assume that we have one feature set and two class labels (0,1)

Feature set = [1, 0, 1, 0], class label = [0, 1, 0, 1]

According to the calculation of homogeneity, completeness, and v-measure, we maintain the following scores:

Homogeneity = 1.0

Completeness = 1.0

V-measure = 1.0

This indicates that it is a perfect score. As seen in the example, the feature set satisfies homogeneity because the same data points are members of a single class, and each class has members of the same data points, which provides perfect completeness. As v-measure is the harmonic mean between homogeneity and completeness. As a result, the score tends to be 1 which indicates a completely perfect labeling.

V-measure Calculation Demonstration 2

Feature set = [0, 1, 2, 3], class label = [0, 0, 1, 1]

According to the calculation of homogeneity, completeness, and v-measure, we maintain the following scores:

Homogeneity = 0.99

Completeness = 0.49

V-measure = 0.66

The scores above show high homogeneity but low completeness and that refers to splitting classes into more clusters.

V-measure Calculation Demonstration 3

Feature set = [0, 1, 0, 1], class label = [0, 0, 1, 1]

According to the calculation of homogeneity, completeness, and v-measure, we maintain the following scores:

Homogeneity = 0.0

Completeness = 0.0

V-measure = 0.0

The scores above shows that these data points are neither homogeneous nor complete due to distribution of the data points among different class labels.

Validity Measure Feature Reduction

For the feature selection technique, we rely on homogeneity and completeness, which is derived for v-measure as defined by Rosenberg and Hirschberg (2007). The calculation of homogeneity and completeness is based on conditional entropy analysis. They measure the clustering quality of the traffic network data by forming data points based on related frequencies of network IP addresses, network ports, and other features. In order to select the suitable features, we intend to implement the K-means clustering algorithm due to its simplicity, easiness in interpreting the clustering results, and efficiency in terms of computational cost. K-means can cluster each attribute separately, and then use the v-measure to assess the clustering quality as follows.

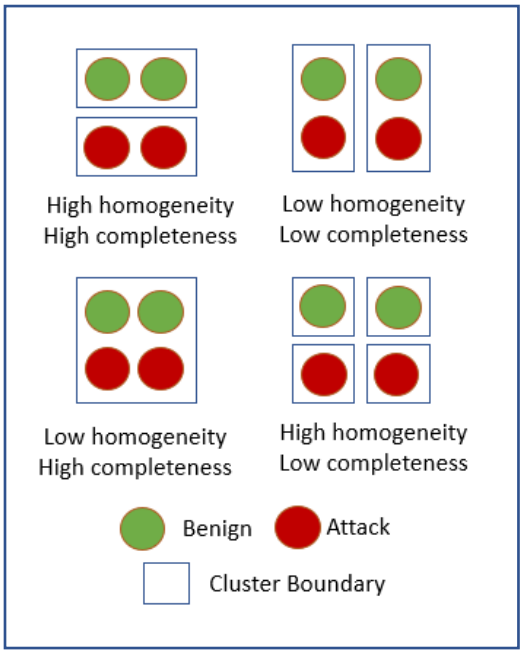
Cluster homogeneity as defined by Han et al. (2011) requires that the purer networks traffic data in a clustering is, the more reliable the cluster is. All data points in a data set, D belong to label classes K_1, K_2 for binary classification. Assume C_1 , a clustering where $C \in C_1$ includes data points from two classes, and C_2 a clustering that corresponds to C_1 except that C_2 is divided into two clusters that contains data points in K_1, K_2 respectively. Therefore, in terms of clustering quality measure Q , C_2 has a homogeneity score higher than C_1 as follows:

$Q\left(C_2, C_g\right) > Q(C_1, C_g)$ (where Cg is a ground truth of clustering C, Q is measure score) and as shown in Figure 5 high homogeneity and low completeness.

Cluster completeness as defined by (Han et al., 2011) requires that the networks traffic data of a singular cluster belong to the same class. Consider clustering C_1, which contains clusters C_1 and C_2 where the members of clustering belong to the same class as indicated by K_1, K_2 . Assume that clustering C_2 is identical to C_1 except that C_1 and C_2 are integrated into one cluster in C_2. Therefore, in terms of clustering quality measure Q, C_2 has a completeness score higher than C_1 as follows: $Q\left(C_2, C_g\right) > Q(C_1, C_g)$ where (Cg is a ground truth of clustering C, Q is measure score) and as shown in Figure 5 low homogeneity and high completeness.

Figure 5

Cluster Outlines of Completeness and Homogeneity Values



Therefore, the clustering is perfectly homogeneous when the data points of each cluster belong to the same class label (benign or attack). However, it is not complete because all data points do not belong to the same class label. Figure 5 shows an extreme example of the cluster boundaries that result in higher and lower completeness and homogeneity values.

Experiments

To demonstrate the effectiveness of using v-measure for feature selection, we have performed five experiments.

Experiment A: After pre-processing and transforming raw data into an understandable format, the three classifiers were applied to all training and testing datasets. Then we applied our new feature selection v-measure. If the v-measure score is greater than 1, completeness is weighted more strongly for the purpose of classification. If v-measure scores are less than 1; we concluded homogeneity is weighted more strongly in the classification process. According to the features that have been selected by v-measure, the same number of features was selected for F-measure, and Information Gain.

Experiment B: F-measure was applied to all training and testing datasets and evaluated the outcomes using the three machine learning algorithms: Decision Tree, Random Forest, and AdaBoost. We conducted a comparison between the v-measure and F-measure, and Information Gain based on the confusion matrices they produced.

Experiment C: We have applied Information Gain (IG) that was developed as a feature selection to all training and testing dataset and testify the outcomes using the three machine learning algorithms: Decision Tree, Random Forest, and AdaBoost. We conducted a comparison between the Information Gain, v-measure, F-measure based on the confusion matrix.

Experiment D: We have applied the three machine learning algorithms: Random Forest, Decision Tree, and AdaBoost. The implementation was based on the identical features between v-measure, F-measure, and Information Gain.

Experiment E: In this experiment, we have applied the three machine learning algorithms upon the top 10 features that have been extracted by v-measure, F-measure, and Information Gain.

Summary

The research methodology deploys the proposed feature selection technique v-measure upon CICIDS-2017 dataset. In this section, the implementation of v-measure was explained in detail with examples its performance improvement of three classifiers has been presented. The experimental analysis demonstrated the importance of the feature dimensionality reduction techniques which directed to better outcomes. Five different experiments have been conducted in order to present the efficiency of the v-measure among other feature selection techniques. Despite the huge number of audits and features, v-measure was able to achieve good performance in terms of dimensionality reduction while maintaining low false positive rate, low false negative rate, and high detection rate.

Chapter 4

Results

The primary objective of this research is to improve the performance of anomaly-based intrusion detection systems and reduce feature dimensionality by adopting a new feature selection metric v-measure. This research utilized v-measure to select the most significant features and provided objective evaluations of three supervised learning algorithms in maintaining high detection and accuracy rates. A detailed analysis of the results of each experiment has been presented along with an evaluation of results in regard to the accomplishments of the research goals.

Experiment Criteria

Five experiments have been conducted to evaluate the implementation of the proposed feature selection technique v-measure. F-measure and Information Gain (IG) have been utilized as feature selection techniques in order to compare their selected features to v-measure's selected features. Three machine learning algorithms have been applied to evaluate the performance of the selected features.

The experiment settings include Anaconda Python Distribution 3.7, Jupyter notebook, and Scikit-learn. All experiments were deployed on an Amazon SageMaker studio https://aws.amazon.com/console/ (Liberty et al., 2020). Amazon SageMaker is a cloud machine-learning platform that supports elastic learning and incremental training. Amazon Elastic Compute Cloud (EC2) provides the ability to choose the configuration and the capacity with minimal friction. They have been widely used as third-party resources and services to reduce the training cost (Qiu et al., 2021; Carmona et al., 2021). The instances that have been utilized are memory optimized instances (ml.r5.2xlarge) to operate fast performance with 8 virtual CPU and

64 GB memory to pre-processing the dataset, selecting the features, and applying the classifiers. Amazon SageMaker platform helps to reduce the computational cost in term of memory and execution time.

Data Preprocessing

As mentioned in the methodology section, in order to compute v-measure, each feature is processed based on one of the two class labels (i.e., benign, attack) of each data point. This has led to a partition of the original dataset into two clusters.

An explanation of the data preprocessing was mentioned in the previous chapter. CICIDS-207 dataset has been prepared and transformed into an understandable format. To perform the preprocessing steps, Python scripts were written using Pandas library for data analysis and DataFrame instances, and Numpy for numerical operations on a large amount of data (McKinney, 2012). The CICIDS-2017 dataset has 2,830,743 records. Class label (begin) which represents 2,273,097 records has been replaced with 0 and 557,646 records which represent (attacks) that have been replaced with 1.

After removing records with missing values which are 1,358 as and/or infinity values which are 1,509 records, 2,827,876 records have been obtained for usage in the next stage. A few empirical studies utilized different data partitions for training and testing such as 70:30, 80:20, and 60:40 (Gholamy et al., 2018; Soni, & Sharma, 2014). The studies resulted in high accuracy rates. Therefore, the dataset has been split randomly with the ratio of 70% for the training and 30% for testing for all experiments. There are some features that represent complicated attacks on modern networks based on their network traffic feature.

Table 2

Features Numbers and Names

F-ID	Feature Name	F-ID	Feature Name	F-ID	Feature Name	F-ID	Feature Name
1	Destination Port	21	Fwd IAT Total	41	Packet Length Mean	61	Bwd Avg Packets/Bulk
2	Flow Duration	22	Fwd IAT Mean	42	Packet Length Std	62	Bwd Avg Bulk Rate
3	Total Fwd Packets	23	Fwd IAT Std	43	Packet Length Variance	63	Sub flow Fwd Packets
4	Total Backward Packets	24	Fwd IAT Max	44	FIN Flag Count	64	Sub flow Fwd Bytes
5	Total Length of Fwd Packets	25	Fwd IAT Min	45	SYN Flag Count	65	Sub flow Bwd Packets
6	Total Length of Bwd Packets	26	Bwd IAT Total	46	RST Flag Count	66	Sub flow Bwd Bytes
7	Fwd Packet Length Max	27	Bwd IAT Mean	47	PSH Flag Count	67	Fwd Init Win bytes
8	Fwd Packet Length Min	28	Bwd IAT Std	48	ACK Flag Count	68	Bwd Init Win bytes
9	Fwd Packet Length Mean	29	Bwd IAT Max	49	URG Flag Count	69	Fwd Act Data Pkts
10	Fwd Packet Length Std	30	Bwd IAT Min	50	CWE Flag Count	70	Fwd Seg Size Min
11	Bwd Packet Length Max	31	Fwd PSH Flags	51	ECE Flag Count	71	Active Mean
12	Bwd Packet Length Min	32	Bwd PSH Flags	52	Down/Up Ratio	72	Active Std
13	Bwd Packet Length Mean	33	Fwd URG Flags	53	Average Packet Size	73	Active Max
14	Bwd Packet Length Std	34	Bwd URG Flags	54	Fwd Segment Size Avg	74	Active Min
15	Flow Bytes/s	35	Fwd Header Length	55	Bwd Segment Size Avg	75	Idle Mean
16	Flow Packets/s	36	Bwd Header Length	56	Fwd Header Length	76	Idle Std
17	Flow IAT Mean	37	Fwd Packets/s	57	Fwd Avg Bytes/Bulk	77	Idle Max
18	Flow IAT Std	38	Bwd Packets/s	58	Fwd Avg Packets/Bulk	78	Idle Min
19	Flow IAT Max	39	Min Packet Length	59	Fwd Avg Bulk Rate	79	Label
20	Flow IAT Min	40	Max Packet Length	60	Bwd Avg Bytes/Bulk		

There are some features that represent complicated attacks on modern networks based on their network traffic feature. For instance, some features are required to identify normal activities such as the Bwd Packet Length Min, and the Fwd Average Package Length features. While some of them are required to detect malicious activities such as Subflow Fwd Bytes, Total Length of Fwd Packets, Init Win Bytes Forward, and Bwd Packet Length Std. Some features can be easily removed due to their less statistical measures values such as Bwd PSHF lags, Fwd URG Flags, Bwd URG Flags, RST Flag Count, CWE Flag Count, ECE Flag Count, Fwd Avg Bytes/Bulk, Fwd Avg Packets/Bulk, Fwd Avg Bulk Rate, Bwd Avg Bytes/Bulk, Bwd Avg Packets/Bulk, and Bwd Avg Bulk Rate. A full list of these features is included in Appendix A. The features in Table 2 are ready to be evaluated by conducting the following five experiments.

Evaluation Criteria

The results were shown on the standard metrics for binary classification of two classes Benign and Attack: true positive (TP), false positive (FP), true negative (TN), and false negative (FN) (Marsland, 2014). The classification accuracy, detection rate, false positive rate, and false negative rate were calculated from these metrics:

1) Accuracy: the ratio of correctly classified samples to the total number of samples.

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

2) Detection Rate (DT): the ratio between total numbers of detected attacks to the total number of attacks in the dataset.

$$\text{DT} = \frac{TP}{TP+TN}$$

3) False Positive Rate (FPR): the number of non-intrusions inaccurately detected; false positive rate is defined as:

$$\text{FPR} = \frac{FP}{FP+TN}$$

4) False Negative Rate (FNR): the number of intrusions inaccurately detected; false negative rate is defined as:

$$\text{FNR} = \frac{FN}{FN+TN}$$

Experiment A: Applying V-measure

After preprocessing the dataset, the three classifiers were run against the training data set to evaluate all 78 features using the classification metric and compare their results with our proposed feature selection technique such as v-measure in terms of data reduction and obtaining better classification metrics.

The same training and testing data was used to provide a comparison to other classifiers Decision Tree, Random Forest, and Adaboost using 78 features the results showed in Table 3.

Table 3

Performance Metric for All Features

Classifiers	Accuracy	Detection Rate	False Positive Rate	False Negative Rate
DT	99.8%	99.6%	0.0007	0.0007
RF	99.8%	99.7%	0.0006	0.0006
AdaBoost	99.2%	98.3%	0.004	0.003

Training the data set using Random Forest provided the best accuracy on average but at a high computational cost.

The model ran for up to 12 hours; however, the other two classifiers Decision Tree and AdaBoost provided lower accuracy scores at a lower computational cost. These two classifiers ran for up to 30 minutes. In terms of memory usage, loading the data and applying the three classifiers consumed over 30 GB as shown in Table 4.

Table 4

Running Time

Classifiers	All Features	44 features by v-measure	37 features by F-measure	35 features by Information Gain
DT	30 minutes	12 minutes	8 minutes	13 minutes
RF	720 minutes	360 minutes	348 minutes	362 minutes
Adaboost	34 minutes	13 minutes	15 minutes	15 minutes

According to the results in Table 4, it seems v-measure with the selected 44 features reduced the running time; similarly, Information gain provided a lower running time compared to all list of features. V-measure, homogeneity, and completeness as shown in figure 6 have been applied to every feature.

Figure 6

V-measure, Homogeneity, and Completeness

$$V_\beta = \frac{(1+\beta) * H * C}{(\beta * H) + C} \quad (1)$$

$$H = \begin{cases} 1 & if\ H(C,K) = 0 \\ 1 - \frac{H(C|K)}{H(C)} & else \end{cases} \quad (2)$$

$$C = \begin{cases} 1 & if\ H(K,C) = 0 \\ 1 - \frac{H(K|C)}{H(K)} & else \end{cases} \quad (3)$$

Confusion matrix metrics have been deployed for the sake of evaluating the performances of the selected features. If the v-measure score is greater than 1, completeness is weighted more strongly for the purpose of classification, if v-measure scores are less than 1; we concluded that homogeneity is weighted more strongly in the classification process. After calculating v-measure score to every feature, what are the features that could be used an input and impact the model

and produce better classification metric. Therefore, a threshold has to be selected. To determine the threshold values utilized as a reference for the selected features in v-measure as well as F-measure, and Information Gain. In this research the threshold value is determined independently by calculating the Mean to ensure selecting the best feature that have high importance. It should be noted that there are alternative approaches in the literature to select significant features based on their rankings. For example, research conducted by Tsai and Sung (2020) computed the average of each frequency to obtain the best features threshold value. In another research study, a variance threshold is used to remove the low variance features. It removes all zero-variance by default as well as the features that have similar values in all datapoints (Fida et al., 2021). Despite different approaches, we do not anticipate significant differences in performance even if we change to a different approach as they are all schemes based on a single threshold value.

After implementing v-measure among these selected features, the v-measure average value totals out to 0.063. These values indicate that perfect labeling is more homogeneous and less complete. The average value of homogeneity is 0.321 as shown in Table 5, while the average value of completeness is 0.141. Consequently, the features have been selected based on the v-measure average value. There have been 44 selected features that have a value of 0.063 or higher as indicated in Table 6.

Table 5

Homogeneity, Completeness, and V-measure for All Features

F-ID	**H**	**C**	**VM**	**F-ID**	**H**	**C**	**VM**
F1	0.587	0.077	0.135	F41	0.670	0.048	0.089
F2	0.643	0.030	0.056	F42	0.718	0.052	0.096
F3	0.165	0.037	0.060	F43	0.714	0.052	0.097
F4	0.254	0.056	0.091	F44	0.028	0.091	0.043
F5	0.575	0.054	0.099	F45	0.009	0.023	0.013
F6	0.599	0.057	0.104	F46	0.000	0.024	0.000
F7	0.497	0.054	0.098	F47	0.030	0.024	0.027

F8	0.233	0.047	0.078	F48	0.015	0.012	0.013
F9	0.443	0.041	0.074	F49	0.031	0.050	0.039
F10	0.400	0.051	0.091	F50	0.000	0.022	0.000
F11	0.526	0.058	0.105	F51	0.000	0.024	0.000
F12	0.223	0.039	0.066	F52	0.024	0.015	0.018
F13	0.576	0.053	0.097	F53	0.725	0.052	0.097
F14	0.390	0.059	0.103	F54	0.443	0.041	0.074
F15	0.767	0.033	0.063	F55	0.576	0.053	0.097
F16	0.709	0.031	0.060	F56	0.348	0.052	0.090
F17	0.658	0.030	0.057	F57	0.000	1.000	0.000
F18	0.555	0.032	0.061	F58	0.000	1.000	0.000
F19	0.576	0.030	0.057	F59	0.000	1.000	0.000
F20	0.158	0.016	0.029	F60	0.000	1.000	0.000
F21	0.446	0.035	0.065	F61	0.000	1.000	0.000
F22	0.490	0.035	0.066	F62	0.000	1.000	0.000
F23	0.433	0.042	0.076	F63	0.165	0.037	0.060
F24	0.465	0.037	0.069	F64	0.575	0.054	0.099
F25	0.136	0.017	0.030	F65	0.254	0.056	0.091
F26	0.422	0.043	0.078	F66	0.599	0.057	0.104
F27	0.468	0.042	0.078	F67	0.540	0.085	0.147
F28	0.378	0.044	0.078	F68	0.504	0.085	0.146
F29	0.434	0.045	0.082	F69	0.114	0.030	0.047
F30	0.218	0.037	0.063	F70	0.085	0.041	0.055
F31	0.009	0.023	0.013	F71	0.282	0.051	0.087
F32	0.000	1.000	0.000	F72	0.035	0.015	0.021
F33	0.000	0.022	0.000	F73	0.281	0.051	0.087
F34	0.000	1.000	0.000	F74	0.275	0.054	0.090
F35	0.348	0.052	0.090	F75	0.243	0.054	0.088
F36	0.356	0.057	0.099	F76	0.047	0.019	0.027
F37	0.696	0.031	0.059	F77	0.249	0.062	0.099
F38	0.620	0.030	0.057	F78	0.258	0.054	0.089
F39	0.231	0.047	0.078	F-ID: Feature ID, H: homogeneity C: completeness, VM: v-measure			
F40	0.528	0.055	0.099				

In this experiment, the three classifiers Decision Tree, Random Forest, and AdaBoost have been implemented to evaluate the performance of the selected feature. Based on the classification confusion matrix, the selected features by v-measure. Table 7 illustrated the performance metrics for the selected features by v-measure.

Table 6

Selected Features by V-measure

F-ID	Homogeneity	Completeness	V-measure	F-ID	Homogeneity	Completeness	V-measure
F15	0.767	0.033	0.063	F56	0.348	0.052	0.090
F30	0.218	0.037	0.063	F10	0.400	0.051	0.091
F21	0.446	0.035	0.065	F4	0.254	0.056	0.091
F22	0.490	0.035	0.066	F65	0.254	0.056	0.091
F12	0.223	0.039	0.066	F42	0.718	0.052	0.096
F24	0.465	0.037	0.069	F13	0.576	0.053	0.097
F9	0.443	0.041	0.074	F55	0.576	0.053	0.097
F54	0.443	0.041	0.074	F53	0.725	0.052	0.097
F23	0.433	0.042	0.076	F43	0.714	0.052	0.097
F26	0.422	0.043	0.078	F7	0.497	0.054	0.098
F27	0.468	0.042	0.078	F36	0.356	0.057	0.099
F8	0.233	0.047	0.078	F40	0.528	0.055	0.099
F28	0.378	0.044	0.078	F77	0.249	0.062	0.099
F39	0.231	0.047	0.078	F5	0.575	0.054	0.099
F29	0.434	0.045	0.082	F64	0.575	0.054	0.099
F71	0.282	0.051	0.087	F14	0.390	0.059	0.103
F73	0.281	0.051	0.087	F66	0.599	0.057	0.104
F75	0.243	0.054	0.088	F6	0.599	0.057	0.104
F78	0.258	0.054	0.089	F11	0.526	0.058	0.105
F41	0.670	0.048	0.089	F1	0.587	0.077	0.135
F74	0.275	0.054	0.090	F68	0.504	0.085	0.146
F35	0.348	0.052	0.090	F67	0.540	0.085	0.147

Table 7

Performance Metric for the 44 Selected Features by V-measure

Classifier	Accuracy	Detection Rate	False Positive Rate	False Negative Rate
Decision Tree	99.8%	99.6%	0.0008	0.0008
Random Forest	99.9%	99.7%	0.0006	0.0005
AdaBoost	99.9%	97.8%	0.007	0.005

In comparison to all results shown in Table 4, we overserved that v-measure reduced the computational complexity and cost of the algorithms while maintaining the highest accuracy and the lowest false negative rate. In terms of running time, the training data set using Random

Forest was conducted in about 6 hours, and about 12 to 15 minutes for the other two classifiers. The memory usage of the three classifiers was reduced from 32 to 16 GB. The reduction of features leads to the improvement in computational cost by 50%, while maintaining better classification evaluation metrics.

Table 8

Performance Metric for the 44 Selected Features by V-measure, F-measure, and IG

SM	V-measure			F-measure			Information Gain		
	DT	RF	AdaBoost	DT	RF	AdaBoost	DT	RF	AdaBoost
AC	99.8%	99.9%	99.9%	98.7%	98.8	96.7%	99.8%	99.8%	99.0%
DR	99.6%	99.7%	97.8%	98.6%	98.6	86.4%	99.6%	99.7%	98.3%
FPR	0.0008	0.0006	0.007	0.011	0.011	0.007	0.0009	0.0007	0.007
FNR	0.0008	0.0005	0.005	0.003	0.003	0.032	0.0008	0.0006	0.003

In addition, to provide an objective comparison, there are 44 features selected by F-measure and Information Gain that have been compared against the performance of the 44 features that have been selected by v-measure, as seen in Table 8. It's been observed that there are considerable benefits by selecting 44 features from the proposed feature selection technique v-measure. By deploying diverse classifiers, Random Forest outperformed Decision Tree and AdaBoost in putting up the highest score in terms of accuracy and Detection rate.

With the reduction of the features by selecting 44 features and by the implementation of the Random Forest, which gained a high accuracy of 99.9%, a low false positive rate of 0.0006, and a low false negative rate 0.0005.

The selected 44 features by v-measure achieved better results compared to the full features set in that are shown in table 3. In addition, v-measure outperformed F-measure and Information Gain. An analysis of figure 7 illustrates a surprisingly high accuracy and detection rate, low false positive rate of 0.0006 and a false negative rate of 0.0005. The performance of v-

measure and information gain were closely related due to both being calculated by comparing the entropy of the data set before and after a transformation.

Figure 7

Performance Metric for the 44 Selected Features by V-measure, F-measure, and IG

Experiment B: Applying F-measure

A similar process was followed as for Experiment A. The parameter that has been applied is weighted as an average parameter which, provides the weighted mean of the F-measure with

weights equal to class probability. The weight is calculated by the proportion of each class's support value, x relative to the sum of all support values weights.

$$W = \sum_{i=1}^{n} = (x_i * w_i)$$

$$W = weighted\ average, n = number\ of\ terms\ to\ be\ averaged,$$
$$w_i = weights\ applied\ to\ x\ values, x_i = data\ values\ to\ be\ averaged$$

This parameter used to calculate metrics for each label, and find their average weighted by computing the number of true instances for each label.

All features were measured by computing F-measure. The average value of F-measure is 0.282. As a result, there were 35 features that have been selected as the significant features as demonstrated in Table 9.

Table 9

Selected Features by F-measure

Feature-ID	F-measure	Feature-ID	F-measure	Feature-ID	F-measure
F21	0.300	F52	0.436	F45	0.698
F24	0.300	F30	0.440	F47	0.708
F6	0.304	F12	0.450	F51	0.715
F11	0.304	F49	0.670	F46	0.715
F66	0.304	F74	0.678	F33	0.715
F25	0.345	F73	0.678	F50	0.715
F69	0.380	F77	0.678	F32	0.716
F39	0.405	F78	0.678	F34	0.716
F29	0.435	F48	0.684	F57	0.716
F26	0.435	F31	0.698	F58	0.716
F59	0.716	F61	0.716	F44	0.752
F60	0.716	F62	0.716		

Table 10 demonstrates the results of different classification evaluation metrics of F-measure as well as its comparison against v-measure and Information Gain with 35 selected features. All F-measure scores happened to be significantly lower than the high v-measure scores achieved. However, v-measure obtained a lower false negative rate of 0.0004.

Table 10

Classification Evaluation Metric of F-measure, V-measure, and Information Gain

SM	F-measure			V-measure			Information Gain		
	DT	RF	AdaBoost	DT	RF	AdaBoost	DT	RF	AdaBoost
AC	98.4%	98.5%	95.5%	99.8%	99.9%	99.0%	99.8%	99.8%	99.0%
DR	97.6%	97.6%	86.0%	99.5%	99.7%	98.5%	99.6%	99.7%	98.1%
FPR	0.012	0.012	0.021	0.0008	0.0007	0.007	0.0008	0.0006	0.007
FNR	0.005	0.005	0.033	0.0008	0.0004	0.003	0.0009	0.0007	0.004

Figure 8

Performance Metric for the Selected 35 Features by F-measure, V-measure, and IG

The Classification Evaluation Metric of F-measure, v-measure, and Information Gain by using different classifiers assures that the implementation of v-measure was good enough to achieve an overall accuracy, detection rate, false positive rate, and false negative rate as shown in figure 9. The running time and the memory usage in this experiment was indistinguishable in comparison to the experiment run on the selection of the 44 features.

It can be clearly indicated that v-measure outperformed F-measure and Information Gain. The highest accuracy recording was implemented with Random Forest as shown in figure 8, we received a soaring percentage of 99.9%, which corresponded to a false positive and false negative rate of 0.0006 and 0.0004 respectively as presented in figure 9. It also defeats F-measure in having a higher detection rate of 99.7%. Decision Tree and Random Forest both provide equal percentages of false positive and false negative rates. It's also clearly noticeable that Random Forest outperforms Decision Tree in having a higher detection rate on the features that are selected by all feature selection techniques.

Experiment C: Applying Information Gain

Information Gain was implemented to all features in order to select the most significant features. Information Gain's parameters are (discrete_features = auto) to identify whether to recognize all features discrete or continuous, (n_neighborsint = 3) which is the number of neighbors to use for continuous variables, and (copy = true) to make a copy of the given data. Information Gain computes the variance of every feature in the context of the class variable.

Table 11

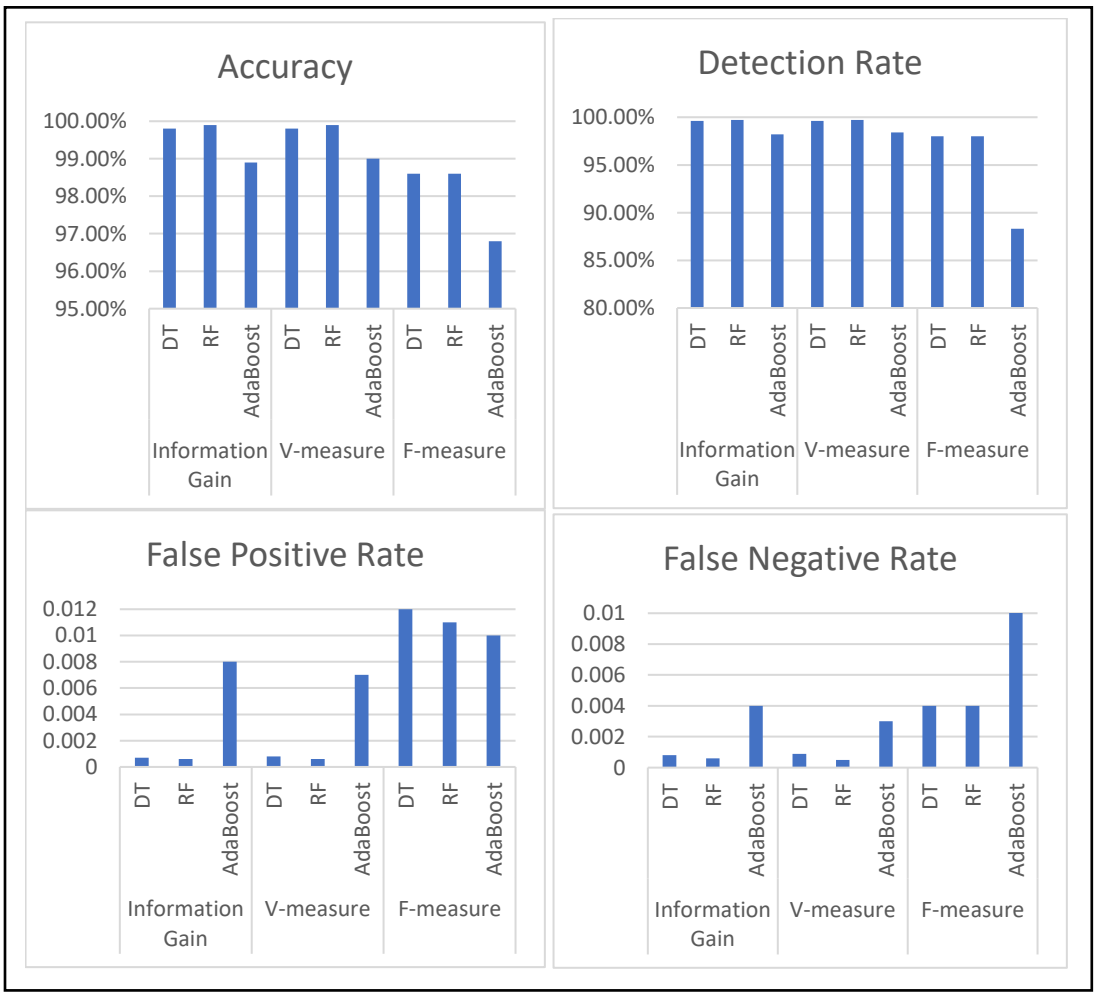
Selected Features by Information Gain

Feature-ID	IG	Feature-ID	IG	Feature-ID	IG
F26	0.149	F16	0.190	F40	0.267
F18	0.150	F24	0.190	F64	0.285

F52	0.150	F14	0.194	F5	0.285
F29	0.161	F38	0.201	F13	0.289
F22	0.169	F2	0.201	F55	0.290
F56	0.172	F15	0.213	F1	0.292
F35	0.173	F19	0.214	F67	0.293
F36	0.177	F54	0.215	F6	0.298
F17	0.177	F9	0.215	F66	0.298
F21	0.179	F7	0.248	F41	0.320
F10	0.181	F68	0.252	F43	0.343
F37	0.189	F11	0.267	F42	0.343
F53	0.347				

Figure 9

Performance Metric for the Selected 37 Features by IG, V-measure, F-measure



The average score was 0.142. As a result, 37 features were chosen to be evaluated by the three machine learning algorithms as shown in Table 11. Table 12 presents the implementation of the three classifiers among 37 features by the results of the classification evaluation metric of all selected features by Information Gain compared to v-measure and F-measure using 37 features confirmed that v-measure achieved a lowest false negative rate.

Referring to the results shown in Table 12. It should be noted that Decision Tree and AdaBoost results were very close. Random Forest achieved a better accuracy for both v-measure and Information Gain. Figures 9 displayed their performance. The running time and the memory usage in this experiment was indistinguishable in comparison to the experiment run on the selection of the 44 and 35 features.

Table 12

Classification Evaluation Metric of Information Gain, V-measure, and F-measure

SM	Information Gain			V-measure			F-measure		
	DT	RF	AdaBoost	DT	RF	AdaBoost	DT	RF	AdaBoost
AC	99.8%	99.9%	98.9%	99.8%	99.9%	99.0%	98.6%	98.6%	96.8%
DR	99.6%	99.7%	98.2%	99.6%	99.7%	98.4%	98.0%	98.0%	88.3%
FPR	0.0007	0.0006	0.008	0.0008	0.0006	0.007	0.012	0.011	0.010
FNR	0.0008	0.0006	0.004	0.0009	0.0005	0.003	0.004	0.004	0.028

Experiment D: Identical Features

Based on our observations, we have concluded that v-measure and F-measure share 14 identical features as shown in Table 13. F-measure and IG have also been concluded to have 8 identical features as shown in Table 14. V-measure and IG have been confirmed to have 29 shared features as shown in Table 15.

The only feature that happened to be below v-measures' average of 0.063 was feature 25 which had an extremely low value of 0.03. While the F-measure features were above the proposed average of 0.282, v-measure still managed to outperform F-measure outstandingly.

Table 13

Identical Features by V-measure and F-measure

Feature ID	Feature Names	Feature value by v-measure	Feature value by F-measure
6	Total Length of Bwd Packets	0.104	0.304
11	Bwd Packet Length Max	0.105	0.304
12	Bwd Packet Length Min	0.066	0.450
21	Fwd IAT Total	0.065	0.300
24	Fwd IAT Max	0.069	0.300
25	Fwd IAT Min	0.030	0.345
29	Bwd IAT Max	0.082	0.435
30	Bwd IAT Min	0.063	0.440
39	Min Packet Length	0.078	0.405
66	Sub flow Bwd Bytes	0.104	0.304
73	Active Max	0.087	0.678
74	Active Min	0.090	0.678
77	Idle Max	0.099	0.678
78	Idle Min	0.089	0.678

Table 14 presents 25 shared identical features between v-measure and Information Gain where all of them maintained above average scores that have been conducted in experiments A and C.

Table 15 is where the 8 identical features between F-measure and Information Gain are showcased. All the features have been proven to be above average as conducted in experiment B and C. Collectively, the three selection feature techniques, v-measure, F-measure, and Information Gain share 7 identical features as shown in Table 16.

Table 14

Identical Features by V-measure and Information Gain

Feature ID	Feature Name	Feature value by v-measure	Feature value by IG
1	Destination Port	0.135	0.191
5	Total Length of Fwd Packets	0.099	0.285
6	Total Length of Bwd Packets	0.104	0.298
7	Fwd Packet Length Max	0.098	0.248
9	Fwd Packet Length Mean	0.074	0.215
10	Fwd Packet Length Std	0.091	0.181
11	Bwd Packet Length Max	0.105	0.267
13	Bwd Packet Length Mean	0.097	0.289
14	Bwd Packet Length Std	0.103	0.194
15	Flow Bytes/s	0.063	0.213
21	Fwd IAT Total	0.065	0.179
22	Fwd IAT Mean	0.066	0.169
24	Fwd IAT Max	0.069	0.190
26	Bwd IAT Total	0.078	0.149
29	Bwd IAT Max	0.082	0.161
35	Fwd Header Length	0.090	0.173
36	Bwd Header Length	0.099	0.177
40	Max Packet Length	0.099	0.267
41	Packet Length Mean	0.089	0.320
42	Packet Length Std	0.096	0.343
43	Packet Length Variance	0.097	0.343
53	Average Packet Size	0.097	0.347
54	Avg Fwd Segment Size	0.074	0.215
55	Avg Bwd Segment Size	0.097	0.290
56	Fwd Header Length	0.090	0.172
64	Sub flow Fwd Bytes	0.099	0.285
66	Sub flow Bwd Bytes	0.104	0.298
67	Init Win bytes forward	0.147	0.293
68	Init Win bytes backward	0.146	0.252

Table 15

Identical Features by F-measure and Information Gain

Feature ID	Feature Name	Feature value by F-measure	Feature value by IG
6	Total Length of Bwd Packets	0.304	0.298
11	Bwd Packet Length Max	0.304	0.267
21	Fwd IAT Total	0.300	0.179
24	Fwd IAT Max	0.300	0.190
26	Bwd IAT Total	0.435	0.149
29	Bwd IAT Max	0.435	0.161

52	Down/Up Ratio	0.436	0.150
66	Sub flow Bwd Bytes	0.304	0.298

Table 16

Identical Features by V-measure, F-measure, and Information Gain

Feature ID	Feature Name	Feature value by v-measure	Feature value by F-measure	Feature value by IG
F6	Total Length of Bwd Packets	0.104	0.304	0.298
F11	Bwd Packet Length Max	0.105	0.304	0.267
F21	Fwd IAT Total	0.065	0.300	0.179
F24	Fwd IAT Max	0.069	0.300	0.190
F26	Bwd IAT Total	0.078	0.435	0.149
F29	Bwd IAT Max	0.082	0.435	0.161
F66	Sub flow Bwd Bytes	0.104	0.304	0.298

Moving on to the next stage, the three classifiers; Decision Tree, Random Forest, and AdaBoost were applied to certain selected features in order to determine which one of the feature selection techniques provided the highest accuracy, detection rate, and the lowest false positive and false negative rate. The results are clearly demonstrated below in Table 17.

Table 17

The Performance of the 7 Identical Features

	Decision Tree	Random Forest	AdaBoost
Accuracy	94.5%	94.6%	91.2%
Detection Rate	88.5%	88.8%	78.2%
False Positive Rate	0.039	0.039	0.054
False Negative Rate	0.028	0.027	0.053

Upon applying the three classifiers among the 7 identical features, the identical features were defeated in terms of all classification performance metrics. These features are discriminative and representative due to the initial windows of the TCP protocol and destination port, and the inter-arrival time. In addition, some features are very important to detect Bot and Infiltration attack types such as Sub flow Bwd Bytes and Total Length of Bwd Packets. While

Total Length of Bwd Packets is required to detect the types of DDoS, DoS Hulk, DoE Golden Eye, and Heartbleed attacks (Stiawan et al., 2020). In addition, the Init Win Fwd Bytes feature is very important to detect the types of Web-Attack, SSH-Patator, and FTP-Patator attacks. While the Min Bwd Package Length feature and Fwd Average Package Length features are needed to detect normal activity (Stiawan et al., 2020).

Experiment E: Top 10 Features

To justify the performance of the proposed feature selection technique, v-measure, the selection of the top 10 features from F-measure, v-measure, and Information Gain have been utilized as a significant part of this experiment.

The results have been compared to the previous conducted experiments. Table 18, 19, and 20 display the feature values accordingly.

Table 18

Top 10 Features by V-measure

Feature ID	Feature Name	Feature value by v-measure
F77	Idle Max	0.099
F5	Total Length of Fwd Packets	0.099
F64	Sub flow Fwd Bytes	0.099
F14	Bwd Packet Length Std	0.103
F66	Sub flow Bwd Bytes	0.104
F6	Total Length of Bwd Packets	0.104
F11	Bwd Packet Length Max	0.105
F1	Destination Port	0.135
F68	Init Win bytes backward	0.146
F67	Init Win bytes forward	0.147

Table 19

Top 10 Features by F-measure

Feature ID	Feature Name	Feature value by F-measure
F50	CWE Flag Count	0.715
F32	Bwd PSH Flags	0.716

F34	Bwd URG Flags	0.716
F57	Fwd Avg Bytes/Bulk	0.716
F58	Fwd Avg Packets/Bulk	0.716
F59	Fwd Avg BulkRate	0.716
F60	Bwd Avg Bytes/Bulk	0.716
F61	Bwd Avg Packets/Bulk	0.716
F62	Bwd Avg Bulk Rate	0.716
F44	FIN Flag Count	0.752

Table 20

Top 10 Features by Information Gain

Feature ID	Feature Name	Feature value by IG
F13	Bwd Packet Length Mean	0.289
F55	Avg Bwd Segment Size	0.290
F1	Destination Port	0.292
F67	Init Win bytes forward	0.293
F6	Total Length of Bwd Packets	0.298
F66	Sub flow Bwd Bytes	0.298
F41	Packet Length Mean	0.320
F43	Packet Length Variance	0.343
F42	Packet Length Std	0.343
F53	Average Packet Size	0.347

After obtaining the top 10 features among the three feature selection techniques, the three classifiers, Decision Tree, Random Forest, and AdaBoost are applied to selected features in order to determine which one of the feature selection techniques provided the highest accuracy, detection rate, as well as a low false positive rate and false negative rate as shown in Table 21.

Table 21

Classification Evaluation Metric of Top 10 by V-measure, F-measure, and IG

SM	V-measure			F-measure			Information Gain		
	DT	RF	AdaBoost	DT	RF	AdaBoost	DT	RF	AdaBoost
AC	99.8%	99.8%	98.3%	80.9%	80.9%	80.9%	99.6%	99.6%	97.9%
DR	99.4%	99.4%	96.1%	10.5%	10.6%	10.4%	98.6	98.6	94.7
FPR	0.0009	0.0009	0.010	0.018	0.018	0.018	0.001	0.001	0.012
FNR	0.001	0.001	0.009	0.182	0.182	0.182	0.003	0.003	0.012

According to the results presented in Table 21, it's obvious that v-measure won the round versus F-measure by selecting the top 10 features that play a major role in determining the performance between both feature selection techniques. The highest detection rate and the lowest false negative rate were obtained by v-measure. The following figures show the performance comparison between v-measure and F-measure.

Figure 10

Performance Metric for the Top 10 Features by V-measure, F-measure, and IG

As shown in Table 21, upon applying the three classifiers on the top 10 features to evaluate their performance with four classification metrics are utilized. The highest accuracy and detection rate, lowest false positive and lowest false negative belong to v-measure. Lastly, after the conduction of five experiments with different parameters and concepts, v-measure has achieved the best results and outperformed all the other feature selection techniques as illustrated in figure 10.

These features are significant due to their impact on determining potential attack such as destination port and some features used to separate appendices from regular flows such as SYN Flag Count, Bwd Pkt Len Mean. Flow construction features such as Sub flow Bwd Bytes and Flow Bytes are important features that must be thoroughly tracked and analyzed in real time. Therefore, it is most definitely important to indicate what includes a single input for the detection system (Engelen et al., 2021).

Overall, this highlights the significance of v-measure and its capabilities to select the considerable features on the CICIDS-2017 dataset. V-measure outperformed F-measure and Information Gain in terms of accuracy, detection rate, false positive rate, and false negative rate. Random Forest defeats Decision Tree and AdaBoost in providing better results in all aspects. Considering the running time, the Decision Tree classifier was the best algorithm with the shortest execution time and highest accuracy. However, Random Forest was the worst in terms of execution time.

The experiments conducted above were capable to demonstrate the improvements of the proposed feature selection v-measure. It was stated that the new feature extraction has selected the best features for the participation in the classification process among F-measure and Information Gain.

Algorithm: V-measured-Based-IDS

Input: Feature set $\boldsymbol{F} = \{\boldsymbol{f_1}, \boldsymbol{f_2}, \dots, \boldsymbol{f_i} \dots, \boldsymbol{f_n}\}$

where $\boldsymbol{1 \le i \le n}$, $\boldsymbol{C}$, $\boldsymbol{K}$, dataset $\boldsymbol{D}$ where $\boldsymbol{|D| = m}$

Output: V-measure of all features in $\boldsymbol{F}$: $\boldsymbol{V(F) = \{V(f_1), V(f_2), \dots, V(f_i) \dots, V(f_n)\}}$

where $\boldsymbol{1 \le i \le n}$

1: For all $(\boldsymbol{1 \le i \le n})$

2: Perform clustering of items in $\boldsymbol{D}$ into $\boldsymbol{K}$ clusters based on feature $\boldsymbol{f_i}$

//Calculate homogeneity

3: Calculate $\boldsymbol{H(C) = -\sum_{c=1}^{|C|} \frac{\sum_{k=1}^{|K|} a_{ck}}{m} \log \frac{\sum_{k=1}^{|K|} a_{ck}}{m}}$

4: Calculate $\boldsymbol{H(C|K) = -\sum_{k=1}^{|K|} \sum_{c=1}^{|C|} \frac{a_{ck}}{m} \log \frac{a_{ck}}{\sum_{c=1}^{|C|} a_{ck}}}$

5: If $\boldsymbol{H(C,K) = 0}$ then $\boldsymbol{H = 1}$

6: else $\boldsymbol{H = 1 - \frac{H(C|K)}{H(C)}}$

//Calculate completeness

7: Calculate $\boldsymbol{H(K) = -\sum_{k=1}^{|K|} \frac{\sum_{c=1}^{|C|} a_{ck}}{m} \log \frac{\sum_{c=1}^{|C|} a_{ck}}{m}}$

8: Calculate $\boldsymbol{H(K|C) = -\sum_{c=1}^{|C|} \sum_{k=1}^{|K|} \frac{a_{ck}}{m} \log \frac{a_{ck}}{\sum_{k=1}^{|K|} a_{ck}}}$

9: If $\boldsymbol{H(K,C) = 0}$ then $\boldsymbol{K = 1}$

10: else $\boldsymbol{K = 1 - \frac{H(K|C)}{H(K)}}$

//Calculate $\boldsymbol{\beta}$

11: Calculate True Positive ($\boldsymbol{TP}$), True Negative ($\boldsymbol{TN}$), False Positive ($\boldsymbol{FP}$), False Negative ($\boldsymbol{FN}$)

12: Calculate $\boldsymbol{\text{precision} = \frac{TP}{TP+FP}}$, $\boldsymbol{\text{recall} = \frac{TP}{TP+FN}}$

13: Calculate $\boldsymbol{\beta = \frac{precision}{recall}}$

//Calculate v-measure

14: Calculate $\boldsymbol{V(f_i) = \frac{(1+\beta)*H*C}{(\beta*H)+C}}$

15: **return** $\boldsymbol{V(F) = \{V(f_1), V(f_2), \dots, V(f_i) \dots, V(f_n)\}}$

Computational Complexity

In order to calculate the computational complexity, let C be the set of classes in the dataset CICIDS2017, K the set of clusters in the dataset. One of the implications of this method is computational complexity. Due to the complexity of machine learning algorithms, the training and testing times are longer. There could be important performance issues caused by v-measure because it acquires additional computing costs to estimate their number of parameters.

Let the state of the features $F = \{f_0, f_1, f_2, \dots, f_{n-1}\}$. To calculate v-measure, we require to calculate the homogeneity and completeness first as follow:

$$H = \begin{cases} 1 & if\ H(C,K) = 0 \\ 1 - \dfrac{H(C|K)}{H(C)} & else \end{cases}$$

Calculating the cluster homogeneity, let C be the set of classes in the dataset CICIDS2017, K the set of clusters in the dataset in the context of this research, K=2 for binary classification K=2 for binary classification, n the total number of elements, and a_{ck} is the number of elements from class C assigned to cluster K. $H(C|K)$ is maximal and equals $H(C)$ when the clustering does not provide new information.

$$H(C|K) = -\sum_{k=1}^{|K|}\sum_{c=1}^{|C|}\frac{a_{ck}}{n}\log\frac{a_{ck}}{\sum_{c=1}^{|C|}a_{ck}}$$

$$H(C) = -\sum_{c=1}^{|C|}\frac{\sum_{k=1}^{|K|}a_{ck}}{n}\log\frac{\sum_{k=1}^{|K|}a_{ck}}{n}$$

Since the class labeled "Benign" and "Attack". To find the features that belong to these two classes, the homogeneity algorithm examines all features of F and count those features that belong to one class either benign or attack. While we calculate $H(C|K)$ we go for all $|K|$ then for each k we have the nested loop for all $|C|$ and within it a third loop form 1 to $|C|$ this gives: Complexity of computing $H(C|K)$: $O(|K||C|^2)$

Complexity of computing $H(C)$: $O(|C||K|)$

Similarly, in order to calculate the cluster completeness, let C be the set of classes in the dataset CICIDS2017, K is the set of clusters in the dataset, n the total number of elements, and a_{ck} be the number of elements from class C assigned to cluster K, as

$$C = \begin{cases} 1 & if\ H(K,C) = 0 \\ 1 - \dfrac{H(K|C)}{H(K)} & else \end{cases}$$

$$Where$$

$$H(K|C) = -\sum_{c=1}^{|C|}\sum_{k=1}^{|K|}\frac{a_{ck}}{N}\log\frac{a_{ck}}{\sum_{k=1}^{|K|}a_{ck}}$$

$$H(K) = -\sum_{k=1}^{|K|}\frac{\sum_{c=1}^{|C|}a_{ck}}{n}\log\frac{\sum_{c=1}^{|C|}a_{ck}}{n}$$

While we calculate $H(K|C)$ we go for all $|C|$ then for each k we have the nested loop for all $|K|$ and within it a third loop form 1 to $|K|$ this gives:

Complexity of computing $H(K|C)$: $O(|C||K|^2)$

Complexity of computing $H(C)$: $O(|K||C|)$

In the algorithm shown above, the calculation of homogeneity has a complexity of $O(|K||C|^2)$. The calculation of completeness has a complexity of $O(|C||K|^2)$. The algorithm has an overall complexity of $O(nm^2|K||C|^2)$ or $O(nm^2|C||K|^2)$ depending on whether $|C|$ or $|K|$ is larger (n: the number of features, m: the number of data items) because the clustering algorithm generally performs with complexity of $O(m^2)$. Since both $|K|$ (number of clusters) and $|C|$ (number of class labels) are constants in this research, the overall complexity can be simplified as $O(nm^2)$. It should be noted that this complexity cannot be underestimated especially with large datasets (i.e., when *m* is large), as demonstrated by the computation times in our

experiments. The computational time is also proportional to the number of features (i.e., n). Some expedition techniques, such as sampling, could be used on smaller datasets (assuming they follow the same distributions as the original datasets) to achieve better performance by reducing the value m. After ranking all features based on their v-measures, the selected feature set is $F' = \{f_{v_1}, f_{v_2}, \dots f_{v_j}, \dots, f_{v_l}\}$, where $v_j \in V$, $1 \leq j \leq l$, and l is the threshold.

Summary

The research methodology introduced a feature selection technique v-measure and tested it on the CICISS-2017 dataset. The experimental analysis has demonstrated the importance of the feature dimensionality reduction techniques which lead to better outcomes. By conducting experiments to validate the generated feature sets using multiple classifiers, Random Forest is capable to detected anomalous activity and improve the detection rate, obtaining lowest false positive rate and false negative rate. Apparently, there is a noticeable improvement in obtaining a high accuracy of 99.9% recorded by Random Forest, the detection rate of 99.7%, low false positive rate of 0.0006, and false negative rate of 0.0004. For the other classifiers, although the performance is not as good as random forest, still validates the performance if v-measure in terms of high accuracy and low false positive/negative rates. Despite the huge number of audits and features, v-measure was able to achieve good performance in terms of dimensionality reduction while maintaining a low false positive rate, low false negative rate, and a high detection rate.

Chapter 5

Conclusions, Implications, Recommendations, and Summary

This chapter presents conclusions, implications, recommendations, and a summary derived from this research which is to study the effect of feature selection in the intrusion detection systems.

Conclusions

Feature selection is one of the most significant procedures of data preprocessing in data analytics. The data with various features has impacted the computational complexity and increase the amount of resource usage. This research provides empirical evidence that support the proposed feature selection technique through the implementation and evaluation of v-measure. The main focus of this research was to answer the two following research questions:

1- Is v-measure a good feature selection technique in improving intrusion detection based on the CIDISD2017 dataset, while maintaining high detection rate and low false positive and false negative rate at the same time?

2- What are the computational costs of v-measure when it is compared to other statistical measures such as F-measure or Information Gain?

Through the experiments, we can conclude that both questions were answered. The CICIDS-2017 contains a huge volume of network traffic that is based on real traces of benign and malicious activities. The results showed that there was a significant improvement by using v-measure, with the highest accuracy rate and detection rate occurred by Random Forest and Decision Tree of 99.9%, and the lowest false positive and false negative rate gained by only Random Forest. Our proposed feature selection technique combined the advantages of using

homogeneity and completeness which evaluates whether all data points with a particular class are put together in a cluster and whether that cluster contains only them. We conducted several experiments to show that our approach could improve the performance of intrusion detection systems.

Implications

This work can be used as a guide for researchers performing feature selection in the domain of intrusion detection systems. Previous studies in the field of feature selection showed that there is a lack in finding the optimal feature selection techniques due to multiple reasons. Over the years, many feature selection techniques have been proposed, with varied level of success. However, no single feature selection technique dominates the others. Our proposed research identifies a novel metric to help with the evaluation on the significance of different features in an intrusion detection dataset. The proposed metric has demonstrated superior performance on a binary intrusion classification problem. It has the potential to be used alone, or combined with other heuristics, in various anomaly detection fields.

Based on the results presented in the previous chapter, it was noticeable that Random Forest performed remarkably well compared to the other machine learning algorithms such as Decision Tree and AdaBoost. Our research utilized v-measure to reduce superfluous data and extract significant features. The results show that v-measure wasn’t only identified to be a part of selecting the important features in the tested dataset, but to also help reduce false positive and false negative rates and improve detection rates as well. We believe this is a contribution to the body of knowledge in anomaly detection.

Recommendations

Feature selection techniques have gained a considerable amount of interest in the intrusion detection systems domain. The results of this dissertation provided further research ideas for the feature selection techniques in intrusion detection systems. Some potential areas to extend this research consists of the following.

- In this research, the classification problem was considered as a binary classification problem. Moving from this concept, multi classification problems could provide precise analysis of each features set that impacts each type of attack.
- A potential variation of this research is to select the features based on the calculation homogeneity and completeness separately and compare the results with the significant features selected by v-measure.
- In this research, the evaluation process was conducted using only three machine learning algorithms. It would be valuable to use more advanced machine learning techniques, for example, deep learning algorithms such as Convolutional Neural Networks (CNNs) or Recurrent Neural Networks (RNNs) (Banerjee et al., 2019).

Summary

One issue to evaluate an anomaly detection system is that the data source contains irrelevant and duplicate input data. Each feature contains data points that represent significant information about the network activities. Some of these audit records may contain missing or infinity values that could potentially lead to misclassification.

To reduce data volume and eliminate irrelevant features for anomaly detection is sometimes important to use of statistics metrics and machine learning algorithms. An inaccurately selected set of features may lead to a significant reduction in the detection rate and low classification accuracy. Therefore, selecting the most significant features aiming to reduce the computing resources and improve the accuracy.

In our research, an external entropy v-measure was adopted as a feature selection technique. V-measure is an entropy-based cluster evaluation measure that provides a measure of the randomness of features and a measure of information acquired by comparing various features. V-measure is based on two concepts, homogeneity and completeness. V-measure presented a solution in terms of evaluating the quality of the clustering.

The main challenge of this research was to reduce the dataset features as much as possible while maintaining low false positive and false negative rates, and high detection rates in binary intrusion classification.

There were five experiments designed in the methodology chapter. The first experiment was conducted to deploy v-measure over all features and then choose the most proper and important feature. In the second and the third experiments, v-measure was compared to F-measure as one of the entropy measures and IG as a filter feature selection technique. The fourth experiment uses identical features, and the fifth uses the top ten features that have been selected by the three feature selection techniques. The results indicated that v-measure was above our expectation and obtained better results in terms of high accuracy, high detection rate, low false positive rate, and false negative rate.

In summary, v-measure has been developed and evaluated as a feature selection technique using the CICIDS-2017 dataset. This dataset was preprocessed and split for training

(70%) and testing (30%). Two other feature selection techniques were also introduced for a comparison against v-measure, those feature selection techniques appear to be F-measure, and Information Gain. In this research, five experiments have been conducted, as well as three machine learning classifiers Decision Tree, Random Forest, and AdaBoost have been applied for an evaluation of which of the three feature selection techniques provided better performance. Based on the experimental results, we claim that v-measure is a viable candidate for feature selection when high dimensional and big volume of data is used for anomaly detection.

Appendices

Appendix A: CICIDS-2017 Features Description (Sharafaldin et al., 2018)

Feature ID	Feature Names	Feature Description
1	Destination Port	Destination Port
2	Flow Duration	Duration of the flow in Microsecond
3	Total Fwd Packets	Total packets in the forward direction
4	Total Backward Packets	Total packets in the backward direction
5	Total Length of Fwd Packets	Total size of packet in forward direction
6	Total Length of Bwd Packets	Total size of packet in backward direction
7	Fwd Packet Length Max	Maximum size of packet in forward direction
8	Fwd Packet Length Min	Minimum size of packet in forward direction
9	Fwd Packet Length Mean	Mean size of packet in forward direction
10	Fwd Packet Length Std	Standard deviation size of packet in forward direction
11	Bwd Packet Length Max	Maximum size of packet in backward direction
12	Bwd Packet Length Min	Minimum size of packet in backward direction
13	Bwd Packet Length Mean	Mean size of packet in backward direction
14	Bwd Packet Length Std	Standard deviation size of packet in backward direction
15	Flow Bytes/s	Number of flow bytes per second
16	Flow Packets/s	Number of flow packets per second
17	Flow IAT Mean	Mean time between two packets sent in the flow
18	Flow IAT Std	Standard deviation time between two packets sent in the flow
19	Flow IAT Max	Maximum time between two packets sent in the flow
20	Flow IAT Min	Minimum time between two packets sent in the flow
21	Fwd IAT Total	Total time between two packets sent in the forward direction
22	Fwd IAT Mean	Mean time between two packets sent in the forward direction
23	Fwd IAT Std	Standard deviation time between two packets sent in the forward direction
24	Fwd IAT Max	Maximum time between two packets sent in the forward direction
25	Fwd IAT Min	Minimum time between two packets sent in the forward direction
26	Bwd IAT Total	Total time between two packets sent in the backward direction
27	Bwd IAT Mean	Mean time between two packets sent in the backward direction
28	Bwd IAT Std	Standard deviation time between two packets sent in the backward direction

29	Bwd IAT Max	Maximum time between two packets sent in the backward direction
30	Bwd IAT Min	Minimum time between two packets sent in the backward direction
31	Fwd PSH Flags	Number of times the PSH flag was set in packets travelling in the forward direction
32	Bwd PSH Flags	Number of times the PSH flag was set in packets travelling in the backward direction (0 for UDP)
33	Fwd URG Flags	Number of times the PSH flag was set in packets travelling in the backward direction (0 for UDP)
34	Bwd URG Flags	Number of times the URG flag was set in packets travelling in the backward direction (0 for UDP)
35	Fwd Header Length	Total bytes used for headers in the forward direction
36	Bwd Header Length	Total bytes used for headers in the backward direction
37	Fwd Packets/s	Number of forward packets per second
38	Bwd Packets/s	Number of backward packets per second
39	Min Packet Length	Min Packet Length
40	Max Packet Length	Max Packet Length
41	Packet Length Mean	Mean length of a packet
42	Packet Length Std	Standard deviation length of a packet
43	Packet Length Variance	Variance length of a packet
44	FIN Flag Count	Number of packets with FIN
45	SYN Flag Count	Number of packets with SYN
46	RST Flag Count	Number of packets with RST
47	PSH Flag Count	Number of packets with PUSH
48	ACK Flag Count	Number of packets with ACK
49	URG Flag Count	Number of packets with URG
50	CWE Flag Count	Number of packets with CWE
51	ECE Flag Count	Number of packets with ECE
52	Down/Up Ratio	Download and upload ratio
53	Average Packet Size	Average size of packet
54	Fwd Segment Size Avg	Average size observed in the forward direction
55	Bwd Segment Size Avg	Average size observed in the backward direction
56	Fwd Header Length	Total bytes used for headers in the forward direction
57	Fwd Avg Bytes/Bulk	Average number of bytes bulk rate in the forward direction
58	Fwd Avg Packets/Bulk	Average number of packets bulk rate in the forward direction
59	Fwd Avg Bulk Rate	Average number of bulk rate in the forward direction

60	Bwd Avg Bytes/Bulk	Average number of bytes bulk rate in the backward direction
61	Bwd Avg Packets/Bulk	Average number of packets bulk rate in the backward direction
62	Bwd Avg Bulk Rate	Average number of bulk rate in the backward direction
63	Sub flow Fwd Packets	The average number of packets in a sub flow in the forward direction
64	Sub flow Fwd Bytes	The average number of bytes in a sub flow in the forward direction
65	Sub flow Bwd Packets	The average number of packets in a sub flow in the backward direction
66	Sub flow Bwd Bytes	The average number of bytes in a sub flow in the backward direction
67	Fwd Init Win bytes	The total number of bytes sent in initial window in the forward direction
68	Bwd Init Win bytes	The total number of bytes sent in initial window in the backward direction
69	Fwd Act Data Pkts	Count of packets with at least 1 byte of TCP data payload in the forward direction
70	Fwd Seg Size Min	Minimum segment size observed in the forward direction
71	Active Mean	Mean time a flow was active before becoming idle
72	Active Std	Standard deviation time a flow was active before becoming idle
73	Active Max	Maximum time a flow was active before becoming idle
74	Active Min	Minimum time a flow was active before becoming idle
75	Idle Mean	Mean time a flow was idle before becoming active
76	Idle Std	Standard deviation time a flow was idle before becoming active
77	Idle Max	Maximum time a flow was idle before becoming active
78	Idle Min	Minimum time a flow was idle before becoming active
79	Label	Class activities label

Appendix B: List of F-measure Features Scores

Feature-ID	F-measure score	Feature-ID	F-measure score
F1	0.001	F40	0.170
F2	0.007	F41	0.000
F3	0.051	F42	0.000
F4	0.245	F43	0.000
F5	0.171	F44	0.752
F6	0.304	F45	0.698
F7	0.171	F46	0.715
F8	0.000	F47	0.708
F9	0.000	F48	0.684
F10	0.000	F49	0.670
F11	0.304	F50	0.715
F12	0.450	F51	0.715
F13	0.000	F52	0.436
F14	0.000	F53	0.000
F15	0.000	F54	0.000
F16	0.000	F55	0.000
F17	0.000	F56	0.001
F18	0.000	F57	0.716
F19	0.007	F58	0.716
F20	0.078	F59	0.716
F21	0.300	F60	0.716
F22	0.000	F61	0.716
F23	0.000	F62	0.716
F24	0.300	F63	0.051
F25	0.345	F64	0.171
F26	0.435	F65	0.245
F27	0.000	F66	0.304
F28	0.000	F67	0.027
F29	0.435	F68	0.069
F30	0.440	F69	0.380
F31	0.698	F70	0.001
F32	0.716	F71	0.000
F33	0.715	F72	0.000
F34	0.716	F73	0.678
F35	0.001	F74	0.678
F36	0.195	F75	0.000
F37	0.000	F76	0.000
F38	0.000	F77	0.678
F39	0.405	F78	0.678

Appendix C: List of Information Gain (IG) Features Scores

Feature-ID	IG score	Feature-ID	IG score
F1	0.292	F40	0.267
F2	0.201	F41	0.320
F3	0.106	F42	0.343
F4	0.141	F43	0.343
F5	0.285	F44	0.014
F6	0.298	F45	0.006
F7	0.248	F46	0.000
F8	0.124	F47	0.051
F9	0.215	F48	0.049
F10	0.181	F49	0.022
F11	0.267	F50	0.000
F12	0.113	F51	0.000
F13	0.289	F52	0.150
F14	0.194	F53	0.347
F15	0.213	F54	0.215
F16	0.190	F55	0.290
F17	0.177	F56	0.172
F18	0.150	F57	0.000
F19	0.214	F58	0.000
F20	0.077	F59	0.000
F21	0.179	F60	0.000
F22	0.169	F61	0.000
F23	0.136	F62	0.000
F24	0.190	F63	0.106
F25	0.064	F64	0.285
F26	0.149	F65	0.141
F27	0.137	F66	0.298
F28	0.087	F67	0.293
F29	0.161	F68	0.252
F30	0.105	F69	0.073
F31	0.006	F70	0.044
F32	0.000	F71	0.126
F33	0.000	F72	0.017
F34	0.000	F73	0.124
F35	0.173	F74	0.125
F36	0.177	F75	0.105
F37	0.189	F76	0.023
F38	0.201	F77	0.112
F39	0.124	F78	0.106

Reference List

Aburomman, A. A., & Reaz, M. B. I. (2016). A novel SVM-kNN-PSO ensemble method for intrusion detection system. *Applied Soft Computing*, *38*, 360-372.

Ahmed, M., Mahmood, A. N., & Hu, J. (2016). A survey of network anomaly detection techniques. *Journal of Network and Computer Applications*, *60*, 19-31.

Aksu, D., Üstebay, S., Aydin, M. A., & Atmaca, T. (2018). Intrusion detection with comparative analysis of supervised learning techniques and fisher score feature selection algorithm. *International Symposium on Computer and Information Sciences* (pp. 141-149). Springer, Cham.

Alharby, A., & Imai, H. (2005). IDS false alarm reduction using continuous and discontinuous patterns. *International Conference on Applied Cryptography and Network Security* (pp. 192-205). Springer, Berlin, Heidelberg.

Anderson, J. P. (1980). Computer security threat monitoring and surveillance. *Technical Report, James P. Anderson Company*.

Amudha, P., & Rauf, H. A. (2011). Performance analysis of data mining approaches in intrusion detection. *International Conference on Process Automation, Control and Computing* (pp. 1-6). IEEE.

Axelsson, S. (2000). The base-rate fallacy and the difficulty of intrusion detection. *ACM Transactions on Information and System Security (TISSEC)*, *3*(3), 186-205.

Aziz, A. S. A., Sanaa, E. L., & Hassanien, A. E. (2017). Comparison of classification techniques applied for network intrusion detection and classification. *Journal of Applied Logic*, *24*, 109-118.

Banerjee, I., Ling, Y., Chen, M. C., Hasan, S. A., Langlotz, C. P., Moradzadeh, N., & Lungren, M. P. (2019). Comparative effectiveness of convolutional neural network (CNN) and recurrent neural network (RNN) architectures for radiology text report classification. *Artificial intelligence in medicine, 97*, 79-88.

Battiti, R. (1994). Using mutual information for selecting features in supervised neural net learning. *IEEE Transactions on neural networks*, *5*(4), 537-550.

Becker, H. (2011). Identification and characterization of events in social media. (Doctoral dissertation, Columbia University).

Belavagi, M. C., & Muniyal, B. (2016). Performance evaluation of supervised machine learning algorithms for intrusion detection. *Procedia Computer Science*, *89*, 117-123.

Carmona, C. U., Aubet, F. X., Flunkert, V., & Gasthaus, J. (2021). Neural contextual anomaly detection for time series. *arXiv preprint arXiv:2107.07702*.

Chandrashekar, G., & Sahin, F. (2014). A survey on feature selection methods. *Computers & Electrical Engineering*, *40*(1), 16-28.

Chauhan, H., Kumar, V., Pundir, S., & Pilli, E. S. (2013). A comparative study of classification techniques for intrusion detection. *International Symposium on Computational and Business Intelligence* (pp. 40-43). IEEE.

Chiu, C. Y., Lee, Y. J., Chang, C. C., Luo, W. Y., & Huang, H. C. (2010). Semi-supervised learning for false alarm reduction. *Industrial conference on data mining* (pp. 595-605). Springer, Berlin, Heidelberg.

Chu, J., Lee, T. H., & Ullah, A. (2020). Component-wise AdaBoost algorithms for high-dimensional binary classification and class probability prediction. In *Handbook of statistics* (Vol. 42, pp. 81-114). Elsevier.

Creech, G., & Hu, J. (2013). A semantic approach to host-based intrusion detection systems using contiguousand discontiguous system call patterns. *IEEE Transactions on Computers*, *63*(4), 807-819.

Cuppens, F., & Miege, A. (2002). Alert correlation in a cooperative intrusion detection framework. *Proceedings 2002 IEEE symposium on security and privacy* (pp. 202- 215). IEEE.

Das, S. (2001). Filters, wrappers and a boosting-based hybrid for feature selection. *Icml* (Vol. 1, pp. 74-81).

Denning, D. E. (1987). An intrusion-detection model. *IEEE Transactions on software engineering*, (2), 222-232.

Dom, B. E. (2002). An information-theoretic external cluster-validity measure. *Proceedings of the Eighteenth conference on Uncertainty in artificial intelligence* (pp.137-145). Morgan Kaufmann Publishers Inc.

Duda, R. O., Hart, P. E., & Stork, D. G. (2012). Pattern classification. *John Wiley & Sons*.

Engelen, G., Rimmer, V., & Joosen, W. (2021). Troubleshooting an intrusion detection dataset: the CICIDS2017 case study. *2021 IEEE Security and Privacy Workshops (SPW)* (pp. 7-12). IEEE.

Fernandes, G., Rodrigues, J. J., Carvalho, L. F., Al-Muhtadi, J. F., & Proença, M. L. (2019). A comprehensive survey on network anomaly detection. *Telecommunication Systems*, *70*(3), 447-489.

Fida, M. A. F. A., Ahmad, T., & Ntahobari, M. (2021). Variance Threshold as Early Screening to Boruta Feature Selection for Intrusion Detection System. In *2021 13th International*

Conference on Information & Communication Technology and System (ICTS) (pp. 46-50). IEEE.

Fleuret, F. (2004). Fast binary feature selection with conditional mutual information. *Journal of Machine learning research, 5*(Nov), 1531-1555.

Furey, T. S., Cristianini, N., Duffy, N., Bednarski, D. W., Schummer, M., & Haussler, D. (2000). Support vector machine classification and validation of cancer tissue samples using microarray expression data. *Bioinformatics, 16*(10), 906-914.

Gholamy, A., Kreinovich, V., & Kosheleva, O. (2018). Why 70/30 or 80/20 relation between training and testing sets: A pedagogical explanation.

Gowadia, V., Farkas, C., & Valtorta, M. (2005). Paid: A probabilistic agent-based intrusion detection system. *Computers & Security, 24*(7), 529-545.

Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of machine learning research, 3*(Mar), 1157-1182.

Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of machine learning research, 3*(Mar), 1157-1182.

Han, J., Kamber, M., & Pei, J. (2011). Data mining concepts and techniques, third edition. *The Morgan Kaufmann Series in Data Management Systems*.

Hand, D., & Christen, P. (2018). A note on using the F-measure for evaluating record linkage algorithms. *Statistics and Computing, 28*(3), 539-547.

Haury, A. C., Gestraud, P., & Vert, J. P. (2011). The influence of feature selection methods on accuracy, stability and interpretability of molecular signatures. *PloS one, 6*(12), e28210.

Hira, Z. M., & Gillies, D. F. (2015). A review of feature selection and feature extraction methods applied on microarray data. *Advances in bioinformatics, 2015*.

Hu, W., Hu, W., & Maybank, S. (2008). Adaboost-based algorithm for network intrusion detection. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, *38*(2), 577-583.

Jha, J., & Ragha, L. (2013). Intrusion detection system using support vector machine. *International Journal of Applied Information Systems (IJAIS)*, *3*, 25-30.

Julisch, K. (2003). Clustering intrusion detection alarms to support root cause analysis. *ACM transactions on information and system security (TISSEC)*, *6*(4), 443-471.

Julisch, K., & Dacier, M. (2002). Mining intrusion detection alarms for actionable knowledge. *ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 366-375).

Khalid, S., Khalil, T., & Nasreen, S. (2014). A survey of feature selection and feature extraction techniques in machine learning. *Science and Information Conference* (pp. 372-378). IEEE.

Khoshgoftaar, T. M., Golawala, M., & Van Hulse, J. (2007). An empirical study of learning from imbalanced data using random forest. *IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2007)* (Vol. 2, pp. 310-317). IEEE.

Kozushko, H. (2003). Intrusion detection: Host-based and network-based intrusion detection systems. *Independent study*.

Lashkari, A. H., Draper-Gil, G., Mamun, M. S. I., & Ghorbani, A. A. (2017). Characterization of Tor Traffic using Time based Features. *ICISSP* (pp. 253-262).

Law, M. H., Figueiredo, M. A., & Jain, A. K. (2004). Simultaneous feature selection and clustering using mixture models. *IEEE transactions on pattern analysis and machine intelligence*, *26*(9), 1154-1166.

Lee, J. H., Lee, J. H., Sohn, S. G., Ryu, J. H., & Chung, T. M. (2008). Effective value of decision tree with KDD 99 intrusion detection datasets for intrusion detection system. *International Conference on Advanced Communication Technology* (Vol. 2, pp. 1170-1175). IEEE.

Li, J., & Liu, H. (2017). Challenges of feature selection for big data analytics. *IEEE Intelligent Systems, 32*(2), 9-15.

Li, J., Cheng, K., Wang, S., Morstatter, F., Trevino, R. P., Tang, J., & Liu, H. (2017). Feature selection: A data perspective. *ACM Computing Surveys (CSUR), 50*(6), 1-45.

Li, W. (2004). Using genetic algorithm for network intrusion detection. *Proceedings of the United States Department of Energy Cyber Security Group, 1*, 1-8.

Liberty, E., Karnin, Z., Xiang, B., Rouesnel, L., Coskun, B., Nallapati, R., ... & Smola, A. (2020). Elastic machine learning algorithms in amazon sagemaker. *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data* (pp. 731-737).

Lima, C. F. L., de Assis, F. M., & de Souza, C. P. (2012). An empirical investigation of attribute selection techniques based on Shannon, Rényi and Tsallis entropies for network intrusion detection. *American Journal of Intelligent Systems*, *2*(5), 111-117.

Lippmann, R. P., Fried, D. J., Graf, I., Haines, J. W., Kendall, K. R., McClung, D., & Zissman, M. A. (2000). Evaluating intrusion detection systems: The 1998 DARPA off-line intrusion detection evaluation. *DARPA Information Survivability Conference and Exposition. DISCEX'00* (Vol. 2, pp. 12-26). IEEE.

Liu, H., & Motoda, H. (2012). *Feature selection for knowledge discovery and data mining* (Vol. 454). Springer Science & Business Media.

Liu, H., Motoda, H., Setiono, R., & Zhao, Z. (2010). Feature selection: An ever evolving frontier in data mining. *Feature selection in data mining* (pp. 4-13). PMLR.

Liu, H., & Lang, B. (2019). Machine learning and deep learning methods for intrusion detection systems: A survey. applied sciences, 9(20), 4396.

Liu, H., & Yu, L. (2005). Toward integrating feature selection algorithms for classification and clustering. *IEEE Transactions on knowledge and data engineering*, *17*(4), 491-502.

Liu, Q., Yin, J., Leung, V. C., Zhai, J. H., Cai, Z., & Lin, J. (2016). Applying a new localized generalization error model to design neural networks trained with extreme learning machine. *Neural Computing and Applications*, *27*(1), 59-66.

Malhi, A., & Gao, R. X. (2004). PCA-based feature selection scheme for machine defect classification. *IEEE Transactions on Instrumentation and Measurement*, *53*(6), 1517-1525.

Manekar, V., & Waghmare, K. (2014). Intrusion detection system using support vector machine (SVM) and particle swarm optimization (PSO). *International Journal of Advanced Computer Research*, *4*(3), 808.

Marsland, S. (2014). *Machine learning: an algorithmic perspective*. Chapman and Hall/CRC.

McHugh, J. (2000). Testing intrusion detection systems: a critique of the 1998 and 1999 darpa intrusion detection system evaluations as performed by lincoln laboratory. *ACM Transactions on Information and System Security (TISSEC)*, *3*(4), 262-294.

Meila, M. (2007). Comparing clusterings—an information-based distance. *Journal of multivariate analysis*, *98*(5), 873-895.

Mishra, P., Varadharajan, V., Tupakula, U., & Pilli, E. S. (2018). A detailed investigation and analysis of using machine learning techniques for intrusion detection. *IEEE Communications Surveys & Tutorials*, *21*(1), 686-728.

Miyahara, K., & Pazzani, M. J. (2000). Collaborative filtering with the simple Bayesian classifier. *International conference on artificial intelligence* (pp. 679-689). Springer, Berlin, Heidelberg.

Moustafa, N., & Slay, J. (2015). UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). *2015 military communications and information systems conference (MilCIS)* (pp. 1-6). IEEE.

Moustafa, N., Slay, J., & Creech, G. (2017). Novel geometric area analysis technique for anomaly detection using trapezoidal area estimation on large-scale networks. *IEEE Transactions on Big Data*, *5*(4), 481-494.

Muna, A. H., Moustafa, N., & Sitnikova, E. (2018). Identification of malicious activities in industrial internet of things based on deep learning models. *Journal of Information Security and Applications*, *41*, 1-11.

Pang, G., Shen, C., Cao, L., & Hengel, A. V. D. (2021). Deep learning for anomaly detection: A review. ACM Computing Surveys (CSUR), 54(2), 1-38.

Patcha, A., & Park, J. M. (2007). An overview of anomaly detection techniques: Existing solutions and latest technological trends. *Computer networks*, *51*(12), 3448-3470.

Parikh, D., & Chen, T. (2008). Data fusion and cost minimization for intrusion detection. *IEEE Transactions on Information Forensics and Security*, *3*(3), 381-389.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., & Vanderplas, J. (2011). Scikit-learn: Machine learning in Python. *Journal of machine learning research*, *12*(Oct), 2825-2830.

Peng, S., Hu, Q., Chen, Y., & Dang, J. (2015). Improved support vector machine algorithm for heterogeneous data. *Pattern Recognition*, *48*(6), 2072-2083.

Peng, X., & Xu, D. (2013). A local information-based feature-selection algorithm for data regression. *Pattern Recognition*, *46*(9), 2519-2530.

Pietraszek, T. (2004). Using adaptive alert classification to reduce false positives in intrusion detection. *International Workshop on Recent Advances in Intrusion Detection* (pp. 102-124). Springer, Berlin, Heidelberg.

Pietraszek, T., & Tanner, A. (2005). Data mining and machine learning—towards reducing false positives in intrusion detection. *Information security technical report*, *10*(3), 169-183.

Qiu, H., Zeng, Y., Guo, S., Zhang, T., Qiu, M., & Thuraisingham, B. (2021). Deepsweep: An evaluation framework for mitigating dnn backdoor attacks using data augmentation. *Proceedings of the 2021 ACM Asia Conference on Computer and Communications Security* (pp. 363-377).

Quinlan, J. R. (1987). Simplifying decision trees. *International journal of man-machine studies*, *27*(3), 221-234.

Rai, K., Devi, M. S., & Guleria, A. (2016). Decision tree based algorithm for intrusion detection. *International Journal of Advanced Networking and Applications*, *7*(4), 2828.

Resende, P. A. A., & Drummond, A. C. (2018). A survey of random forest-based methods for intrusion detection systems. *ACM Computing Surveys (CSUR)*, *51*(3), 48.

Reunanen, J. (2003). Overfitting in making comparisons between variable selection methods. *Journal of Machine Learning Research*, 1371-1382.

Robinson, R. R., & Thomas, C. (2015). Ranking of machine learning algorithms based on the performance in classifying ddos attacks. *Recent Advances in Intelligent Computational Systems (RAICS)* (pp. 185-190). IEEE.

Rosenberg, A., & Hirschberg, J. (2007). V-measure: A conditional entropy-based external cluster evaluation measure. C*onference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL)* (pp. 410-420).

Roth, V., & Lange, T. (2003). Feature selection in clustering problems. *Advances in neural information processing systems*, *16*, 473-480.

Roth, A., Khattra, J., Yap, D., Wan, A., Laks, E., Biele, J., & Shah, S. P. (2014). PyClone: statistical inference of clonal population structure in cancer. *Nature methods*, *11*(4), 396.

Stein, G., Chen, B., Wu, A. S., & Hua, K. A. (2005). Decision tree classifier for network intrusion detection with GA-based feature selection. *43rd annual Southeast regional conference-Volume 2* (pp. 136-141). ACM.

Sharafaldin, I., Lashkari, A. H., & Ghorbani, A. A. (2018). Toward generating a new intrusion detection dataset and intrusion traffic characterization. In *ICISSP* (pp. 108-116).

Shone, N., Ngoc, T. N., Phai, V. D., & Shi, Q. (2018). A deep learning approach to network intrusion detection. *IEEE Transactions on Emerging Topics in Computational Intelligence*, *2*(1), 41-50.

Sindhu, S. S. S., Geetha, S., & Kannan, A. (2012). Decision tree based light weight intrusion detection using a wrapper approach. *Expert Systems with applications*, *39*(1), 129-141.

Singh, B. K., Verma, K., & Thoke, A. S. (2015). Investigations on impact of feature normalization techniques on classifier's performance in breast tumor classification. *International Journal of Computer Applications*, *116*(19).

Solorio-Fernández, S., Carrasco-Ochoa, J. A., & Martínez-Trinidad, J. F. (2019). A review of unsupervised feature selection method. *Artificial Intelligence Review*, 1-42.

Song, Y. Y., & Ying, L. U. (2015). Decision tree methods: applications for classification and prediction. *Shanghai archives of psychiatry*, *27*(2), 130.

Soni, P., & Sharma, P. (2014). An intrusion detection system based on KDD-99 data using data mining techniques and feature selection. *International Journal of Soft Computing and Engineering (IJSCE)*, *4*(3).

Sotiris, V. A., Peter, W. T., & Pecht, M. G. (2010). Anomaly detection through a bayesian support vector machine. *IEEE Transactions on Reliability*, *59*(2), 277-286.

Srihari, V., & Anitha, R. (2014). DDoS detection system using wavelet features and semi-supervised learning. *International Symposium on Security in Computing and Communication* (pp. 291-303). Springer, Berlin, Heidelberg.

Spathoulas, G. P., & Katsikas, S. K. (2010). Reducing false positives in intrusion detection systems. *computers & security*, *29*(1), 35-44.

Stiawan, D., Idris, M. Y. B., Bamhdi, A. M., & Budiarto, R. (2020). CICIDS-2017 dataset feature analysis with information gain for anomaly detection. *IEEE Access*, *8*, 132911-132921.

Sung, A. H., & Mukkamala, S. (2003). Identifying important features for intrusion detection using support vector machines and neural networks. *Symposium on Applications and the Internet. Proceedings.* (pp. 209-216). IEEE.

Tavallaee, M., Bagheri, E., Lu, W., & Ghorbani, A. A. (2009). A detailed analysis of the KDD CUP 99 data set. *IEEE symposium on computational intelligence for security and defense applications* (pp. 1-6). IEEE.

Thomas, G., & Peter, A. F. (2001). Weighted Bayesian Classification based on Support Vector Machine. *18th International Conference on Machine Learning* (pp. 207-209).

Tsai, C. F., & Sung, Y. T. (2020). Ensemble feature selection in high dimension, low sample size datasets: Parallel and serial combination approaches. *Knowledge-Based Systems*, *203*, 106097.

Van Rijsbergen, C. J. (1979). Information Retrieval 2nd Edition Butterworths. *London available on internet*.

Velayutham, C., & Thangavel, K. (2012). A novel entropy based unsupervised feature selection algorithm using rough set theory. *International Conference on Advances in Engineering, Science and Management (ICAESM-2012)* (pp. 156-161). IEEE.

Watson, Gavin. (2018) A comparison of header and deep packet features when detecting network intrusions.

Wang, W., He, Y., Liu, J., & Gombault, S. (2015). Constructing important features from massive network traffic for lightweight intrusion detection. *IET Information Security*, *9*(6), 374-379.

Witten, I. H., Frank, E., Hall, M. A., & Pal, C. J. (2016). *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann.

Xia, D. X., Yang, S. H., & Li, C. G. (2010). Intrusion detection system based on principal component analysis and grey neural networks. *Second International Conference on*

Networks Security, Wireless Communications and Trusted Computing (Vol. 2, pp. 142-145). IEEE.

Xu, D., & Ning, P. (2006). *Correlation analysis of intrusion alerts*. North Carolina State University.

Yao, Y., Su, L., Zhang, C., Lu, Z., & Liu, B. (2019). Marrying Graph Kernel with Deep Neural Network: A Case Study for Network Anomaly Detection. *International Conference on Computational Science* (pp. 102-115). Springer, Cham.

Yin, J., & Wang, J. (2014). A dirichlet multinomial mixture model-based approach for short text clustering. *20th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 233-242). ACM.

Zhang, F., & Wang, D. (2013). An effective feature selection approach for network intrusion detection. *IEEE Eighth International Conference on Networking, Architecture and Storage* (pp. 307-311). IEEE.

Zhang, K., Zhao, F., Luo, S., Xin, Y., & Zhu, H. (2019). An Intrusion Action-Based IDS Alert Correlation Analysis and Prediction Framework. *IEEE Access*, *7*, 150540-150551.

Zhao, Y., & Karypis, G. (2001). Criterion functions for document clustering: Experiments and analysis.

Zhang, Y., & Lee, W. (2000). Intrusion detection in wireless ad-hoc networks. *6th annual international conference on Mobile computing and networking* (pp. 275-283). ACM.

Zuech, R., Khoshgoftaar, T. M., & Wald, R. (2015). Intrusion detection and big heterogeneous data: a survey. *Journal of Big Data*, *2*(1), 3.