Feature Selection for Anomaly–Based Network Intrusion Detection Using Cluster Validity Indices

Tyrone Naidoo*, Jules–Raymond Tapamo†, Andre McDonald*
*Modelling and Digital Science, Council for Scientific and Industrial Research, South Africa
[1]tnaidoo2@csir.co.za
[3]amcdonald@csir.co.za
†College of Agriculture, Engineering and Science, University of Kwazulu–Natal, South Africa
[2]tapamoj@ukzn.ac.za

***Abstract*—A feature selection algorithm that is novel in the context of anomaly–based network intrusion detection is proposed in this paper. The distinguishing factor of the proposed feature selection algorithm is its complete lack of dependency on labelled data, which is rarely available in operational networks. It uses normalized cluster validity indices as an objective function that is optimized over the search space of candidate feature subsets via a genetic algorithm. Feature sets produced by the algorithm are shown to improve the classification performance of an anomaly–based network intrusion detection system over the NSL-KDD dataset. The system approaches the performance attained by using feature sets derived from labelled training data via existing wrapper and filter–based feature selection algorithms.**

***Index Terms*—Network intrusion detection, anomaly detection, feature selection, unsupervised, KDD dataset.**

I. Introduction

In recent years, there has been a rapid increase in Internet usage, which has in turn led to a rise in malicious network activity. Network Intrusion Detection Systems (NIDS) are tools that monitor network traffic with the purpose of rapidly and accurately detecting malicious activity [1]. These systems provide a time window for responding to emerging threats and attacks aimed at exploiting vulnerabilities that arise from issues such as misconfigured firewalls and outdated software.

NIDS are typically classified as misuse–based or anomaly–based systems [2], [3]. Misuse–based systems monitor network traffic for predefined patterns that characterise particular threats [4], [5]. These patterns are generally defined by a security expert only after the threat has been observed; hence, misuse–based systems are unable to reliably detect novel threats. Despite this drawback, misuse–based systems are widely used in practice, due in part to their effectiveness in detecting known threats (as defined with respect to true positive and false positive rates).

Anomaly–based systems construct a profile of legitimate or normal traffic patterns, and monitor network traffic for deviations from the profile, which are subsequently classified as threats or intrusions [2], [6]. While these systems typically exhibit higher false positive rates than misuse–based systems, they are under certain conditions able to detect novel or emerging threats to a network. Due to this advantage, anomaly–based systems have received considerable attention in the research literature.

Several anomaly-based detection methods proposed in the literature use machine learning techniques to construct profiles of legitimate network traffic [6], [7]. In unsupervised anomaly detection methods, feature vectors are constructed from information contained in individual network packets, and an unsupervised learning algorithm is typically applied to identify clusters of samples in the observed data [3], [8]. Under the assumption that the majority of observed network packets are legitimate, the larger clusters are typically labelled as containing legitimate traffic, and outlying data samples are labelled as intrusions.

Feature selection is, in general, an important step in the preprocessing of data for machine learning applications. Due to the richness of information contained in network traffic, it is often possible to construct large feature vectors from network packets, and as such the question of feature selection requires particular attention in the context of network intrusion detection [9]. Previous approaches have performed feature selection via optimization techniques, using the classification accuracy of the NIDS on a subset of the data as an objective function that is to be maximized [10], [11]. While this approach has been shown to improve the performance of the system, it is unrealistic to assume that labelled training data is available in operational networks, which precludes the use of classification accuracy as an objective function in a practical system.

In this paper, a feature selection algorithm that is novel in the context of anomaly–based network intrusion detection is proposed. It uses normalized cluster validity indices as an objective function that is optimized over the search space of candidate feature subsets via a genetic algorithm. The distinguishing factor of this approach to feature selection is its complete lack of dependency on labelled data. Feature sets produced by the algorithm are shown to improve the classification performance of an anomaly–based network intrusion detection system over the NSL-KDD dataset [12]. Despite not requiring access to labelled data, the classification performance of the proposed system approaches the performance attained with effective feature sets that were derived using labelled training data.

The remainder of this paper is set out as follows. In section II, a review of existing anomaly detection methods based on clustering, as used in the context of network intrusion detection, is provided. This review also includes an overview of previous work on feature selection and cluster validity indices. The proposed feature selection algorithm and classifier are presented in section III. The experimental setup is presented in section IV, and the results are presented and compared to the performance attained by using previous feature sets in section V. Conclusions are drawn in section VI.

II. Related Work

This section provides an overview of clustering algorithms relevant to the domain of anomaly–based NIDS, feature selection algorithms for NIDS in general, as well as cluster validity indices.

A. Clustering algorithms for anomaly–based NIDS

Various clustering algorithms have been successfully deployed in the context of anomaly–based network intrusion detection in the literature. These include distance–based clustering algorithms such as single–linkage clustering [8], k–means and k–medoids [13], density–based clustering involving Gaussian mixture models (GMMs) and the expectation–maximization (EM) algorithm [13], [14], and algorithms that combine clustering with some form of outlier detection [15], [16]. Syarif et. al. [13] implemented several of these clustering algorithms for anomaly detection in the NSL-KDD dataset [12], and compared the classification accuracy of these algorithms to that of a misuse-based detection system that makes use of supervised machine learning techniques.

Portnoy et al. [8] performed unsupervised network anomaly detection on the KDD cup 1999 dataset [17] using a variant of single–link clustering. Under the assumption that legitimate network traffic is more prevalent in the dataset than anomalous traffic, a specified fraction of the largest clusters were labelled as legitimate traffic, while the remaining clusters were labelled as intrusions.

An anomaly–based network intrusion detection algorithm that uses outlier detection and k–means clustering was introduced in [15]. The dataset is first clustered using k–means, and outlier scores are computed for each data sample based on its nearest neighbour density and its proximity to each cluster. Data samples with outlier scores exceeding a threshold value are labelled as intrusions.

B. Feature selection algorithms for NIDS

Network traffic is rich in contextual information and as such it is possible to construct high–dimensional feature spaces for machine learning in anomaly–based network intrusion detection [9]. While large feature sets often contain useful information, there may exist a degree of redundancy in certain features, while some features may prove unnecessary for the purpose of detecting certain threats. Feature selection is used to remove unnecessary or redundant features in a feature space, which has the potential to improve the performance of clustering algorithms and as a result the classification accuracy.

Feature selection techniques in general can be divided into two categories, namely filter techniques and wrapper techniques [2]. With filter techniques, features are selected based on their relevance, which is quantified using some form of statistical measure, such as information gain and degree of correlation between the feature and the class label. While filter techniques are independent from the learning algorithm, these algorithms typically require access to labelled data to compute the measure.

Wrapper techniques [18] use the performance of a machine learning algorithm, when applied to samples over a candidate feature subset, as a measure of the relevance of the feature subset. Feature selection is carried out based on this measure; typically, predictive accuracy is used as the measure. In this case, the feature selection algorithm is dependent on the availability of labelled data.

Unless stated otherwise, all of the feature selection algorithms described in the remainder of this section were applied to identify feature subsets in the KDD Cup 1999 dataset [17].

1) Filter–based feature selection: Amiri et al. [19] proposed three feature selection algorithms that are based on maximizing the mutual information and correlation coefficients between features and class labels as measures. The proposed iterative algorithms perform greedy selection of features, in which the best remaining feature with respect to the measures is selected during each iteration. The authors used the feature selection algorithm in the context of misuse detection via a support vector machine (SVM).

Zargari et al. [20] proposed two feature selection algorithms. In the first algorithm, feature subsets are selected based on correlation coefficients, where ”better” subsets have features exhibiting higher degrees of correlation with class labels and lower degrees of correlation with each other, using a greedy algorithm to traverse the search space. The second algorithm uses information gain as a measure of feature relevance, and identifies feature subsets based on the ranking of individual features. Feature sets obtained from these algorithms were used to perform anomaly detection via a random forest algorithm; the proposed system was reported to outperform a feature set constructed through a majority vote of the feature sets from related works [19], [21]–[24].

2) Wrapper–based feature selection: Li et al. [10] proposed a gradual feature removal (GFR) method that was applied to network intrusion detection. The method performs feature selection using the averaged Matthews correlation coefficient (MCC) as a measure of the relevance of a candidate feature subset, as calculated after classification of a subset of data over the candidate feature space using an SVM. Starting with the full feature set, the proposed algorithm iteratively removes the least relevant features from the set, until only one feature remains.

The GFR method proposed by Li et al. [10] was compared against three related feature selection methods, namely the feature removal method (FRM), the sole feature method (SFM), and a hybrid of these methods. The FRM is related to the GFR method in that the ranking of the candidate features for removal in only the first iteration of the GFR method is used. The SFM ranks features based on the average MCC obtained from performing classification on only a single feature at a time. The authors showed that the classification accuracy attained by using the feature set from the GFR method is an improvement over using the FRM and SFM methods.

Dastanpour et. al. [11] proposed an algorithm for wrapper–based feature selection that uses a genetic algorithm (GA) to perform a search over the space of candidate feature subsets. The classification performance of an SVM, as applied to a subset of the data over each candidate feature space, is used as the objective function.

C. Cluster validity indices

Cluster Validity Indices (CVIs) are measures of how well a clustering algorithm manages to identify and assign clusters in a dataset. That is, CVIs are measures of the quality of a clustering result. CVIs are typically defined on the criteria of the compactness and separation of clusters in the feature space [25]. Relative CVIs are often applied in the context of parameter optimization for clustering algorithms (which includes the selection of the number of clusters), in which suitable parameter values are associated with higher cluster validity scores [26].

A number of relative CVIs have been proposed in the literature [26]. The Davies–Bouldin (DB) index [27] is defined as the average ratio of compactness and separation between pairs of clusters. Compactness is defined as the average distance between samples in a cluster and the centroid of cluster, whereas separation is defined as the distance between two cluster centroids.

The Calinski–Harabasz (CH) index [28] is defined as the ratio of the average between–cluster distance (a measure of separation), and the average sum of squared distances of those samples belonging to each separate cluster (a measure of compactness). A comparison of 30 indices performed by Milligan and Cooper [26] showed the CH index as one of the top performing indices in correctly identifying the true number of clusters of synthetic datasets.

III. Proposed System

In what follows, the proposed feature selection algorithm, as well as the proposed classifier, are presented.

A. Feature selection algorithm

The concept behind the proposed feature selection algorithm is that higher cluster validity indices, as individually computed after clustering of a dataset over a population of candidate feature subsets, are indicative of more relevant candidate feature subsets. In turn, these candidate feature subsets translate into improved classifier performance. Note that the computation of the CVIs does not require labelled samples (i.e. it allows for the construction of an unsupervised feature selection algorithm).

A block diagram of the proposed feature selection algorithm is presented in fig. 1.

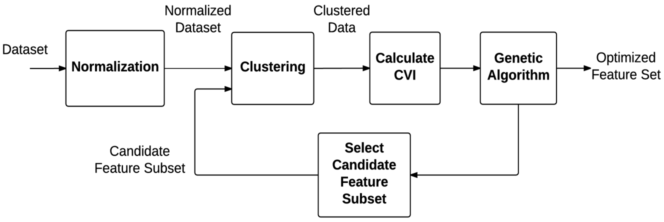

Fig. 1. The proposed feature selection algorithm.

The proposed feature selection algorithm uses a genetic algorithm to search for candidate feature subsets as to minimize the Davies-Bouldin (DB) CVI as the objective function. The CVI is calculated after performing k–means clustering on the dataset of interest, over each candidate feature subset. Each block of the proposed algorithm is described in what follows.

1) Normalization: Feature normalization is performed in order to avoid bias towards features with a broader range of values during the clustering step. Numeric features are normalized using a statistical approach according to the expression

$$\hat{x}_i^{(j)} = (x_i^{(j)} - \mu^{(j)})/(\sigma^{(j)}), \tag{1}$$

where $\hat{x}_i^{(j)}$ is the normalized value of feature j of data sample i, $x_i^{(j)}$ is its original value, and where $\mu^{(j)}$ and $\sigma^{(j)}$ are the mean and standard deviation, calculated over all data samples, of feature j.

Categorical features are normalized using a form of frequency normalization [29], in which the total number of occurrences of each category is divided by the total number of data samples; each occurrence of a category is subsequently replaced by the normalized count corresponding to that category.

2) Clustering: Clustering of the normalized data samples is performed over a candidate subset of features, as selected by the genetic algorithm. The k-means clustering algorithm [30] is used due to its property of convergence, and its computational feasibility over larger datasets [2]. The clustering algorithm is terminated once the iterative assignment of data samples to clusters ceases to change.

The k–means algorithm selects random samples in the data set as initial cluster centroids. As the k–means algorithm is sensitive to the choice of these initial values, k–means clustering is repeated C_R times for each candidate feature subset, where the value of C_R is chosen sufficiently large as to increase the likelihood of finding a high quality clustering result. The k–means algorithm is also repeated over a range of values for the number of clusters C_N to use in k–means clustering, producing a total of C_R clustering results for each choice of C_N.

3) Cluster validity index calculation: The cluster validity index used in this research is a normalized version of the Davies–Bouldin (DB) cluster validity index (refer to section

II-C). The DB CVI incorporates the average Euclidean distance between samples in a cluster and the centroid of the cluster (a measure of compactness), as well as the pairwise Euclidean distance between cluster centroids (a measure of separation).

The calculation of the DB index involves a similarity measure $R_{i,j}$ between each pair of clusters c_i and c_j, which is defined as

$$R_{i,j} = \frac{s_i + s_j}{d_{i,j}}, \tag{2}$$

where s_i and s_j are the compactness measures of clusters c_i and c_j, and $d_{i,j} \triangleq d(v_i, v_j)$ is the Euclidean distance between the centroids v_i and v_j of clusters c_i and c_j. The compactness measure s_i for cluster c_i is defined as

$$s_i = \frac{1}{|c_i|} \sum_{x \in c_i} d(x, v_i), \tag{3}$$

where $|c_i|$ denotes the number of samples in cluster c_i.

The DB index, as calculated for a single clustering result p in which C_N clusters were used ($p = 1, 2, \ldots C_R$) over a feature set F, is defined as

$$\mathrm{DB}(p, C_N, F) = \frac{1}{C_N} \sum_{i=1}^{C_N} R_i, \tag{4}$$

where R_i, $i = 1, \ldots, C_N$, is defined as

$$R_i = \max_{j=1,\ldots,C_N, i \neq j} R_{i,j} \tag{5}$$

Note that the DB index is defined in such a manner that lower values of the index are associated with better clustering results.

The original DB index, as defined in eq. 4, is biased towards smaller feature spaces if the index is to be considered as a measure for comparing clustering performance over feature spaces with different cardinality (this is due to the fewer number of dimensions over which the Euclidean distance is calculated). Handl et al. [31] considered two approaches towards using the feature cardinality to counterbalance this bias. The first approach is to minimize the DB index value while maximizing the number of features, whereas the second approach is to normalize the DB index value using the number of features, thus reversing the bias of the DB index to favour higher dimensions (the new bias can be addressed by multi-objective optimization). The latter approach was followed in this research.

The normalized DB index value, NDB, is defined as

$$\mathrm{NDB}(p, C_N, F) = \frac{1}{|F|} \mathrm{DB}(p, C_N, F) \tag{6}$$

where $|F|$ denotes the cardinality (or number of features) in the candidate feature subset F that was used during clustering.

With reference to fig. 1, the normalized DB index value $\mathrm{NDB}(p, C_N, F)$ is computed for each of the C_R clustering results in which C_N clusters were used. The average value of the index values pertaining to the use of C_N clusters, $\mathrm{NDB}(C_N, F)$, is subsequently computed as

$$\mathrm{NDB}(C_N, F) = \frac{1}{C_R} \sum_{p=1}^{C_R} \mathrm{NDB}(p, C_N, F). \tag{7}$$

4) Genetic Algorithm: Genetic Algorithms (GAs) are heuristic search algorithms that are based on principles of evolution and natural selection [2]. GAs were selected for use in the feature selection algorithm due to their property of exploring a relatively wide region of the solution space [32].

Two configurations of the GA were considered in this paper. In the first configuration, a GA is used to search for feature subsets F to minimize the objective function $\mathrm{NDB}(C_N, F)$ of eq. 7, with a specified number of clusters C_N that remains fixed during optimization. In the second configuration, a multiobjective GA is used to optimize two objective functions. The first objective function is the normalized DB index $\mathrm{NDB}(C_N, F)$ of eq. 7, as was the case in the first configuration. The second objective function is the normalized cardinality of the candidate feature set F, given by $|F|/T_F$, where T_F is the total number of features in the original, full feature set. The specified number of clusters C_N remains fixed during optimization.

The motivation behind the second configuration is from an observation by Handl et al. [31], in that the normalized DB index of eq. 7 is biased towards larger feature spaces. To counterbalance this bias, multiobjective optimization is performed in the second configuration to minimize both the normalized DB index and the number of features in the candidate feature set.

B. Classification

Classification is performed on the NSL-KDD dataset using the optimized feature set returned by the GA. The normalization and clustering steps of the classifier remains the same as in the feature selection algorithm (refer to fig. 1 and sections III-A1 and III-A2). In particular, clustering is again performed over a range of values for the number of clusters C_N. Clusters are labelled based on the assumption that legitimate traffic is more prevalent than traffic pertaining to intrusions. Under this assumption, the proposed classifier labels the M largest clusters of the dataset as containing legitimate traffic samples, while the remaining clusters are labelled as containing intrusions, similar to what was done in [8]. The value of M is varied from 1 to $C_N - 1$ for each selected value of C_N, thereby obtaining a set of $C_N - 1$ possible label assignments for each choice of the number of clusters C_N that are used during the k–means algorithm.

IV. Experimental Setup

A. Dataset

The NSL-KDD dataset [12] was used to evaluate the performance of the proposed feature selection algorithm. This dataset was derived from data packets contained in the KDD Cup 1999 dataset [17], which were captured in a simulated network environment; this network served as a test bed for initiating and evaluating the impact of various types of network intrusions. Several variants of the NSL-KDD datasets are provided; in this research, the 20% training data subset was used. The contents of the dataset are given in table I — due to space limitations, the interested reader is referred to [12] for details on the intrusion classes present in the dataset. A standard set of features constructed for this dataset is listed in table II (refer to [9] for further details on the features).

TABLE I
CONTENTS OF THE 20% NSL-KDD TRAINING DATA SUBSET

Class	# Records	% Records
Denial of service (intrusion)	9 234	36.65%
Probe (intrusion)	2 289	9.09%
Remote to local (intrusion)	209	0.83%
User to root (intrusion)	11	0.04%
Legitimate	13 449	53.39%
Total	**25 192**	**100%**

TABLE II
FEATURES OF THE NSL-KDD DATASET

1: duration	2: prot_type	3: service
4: flag	5: src_bytes	6: dst_bytes
7: land	8: wrong_frag	9: urgent
10: hot	11: num_fail_logins	12: logged_in
13: num_comprom.	14: root_shell	15: su_attempted
16: num_root	17: num_file_cr.	18: num_shells
19: num_acc_files	20: num_outb_cmds	21: is_host_login
22: is_guest_login	23: count	24: srv_count
25: serror_rt	26: srv_serror_rt	27: rerror_rt
28: srv_rerror_rt	29: same_srv_rt	30: diff_srv_rt
31: srv_diff_host_rt	32: dst_host_count	33: dst_host_srv_count
34: dst_hst_smsrv_rt	35: dst_hst_diffsrv_rt	36: dst_hst_smsrc_prtrt
37: dst_hst_srv_dhstrt	38: dst_hst_serror_rt	39: dst_hst_srv_serr_rt
40: dst_hst_rerr_rt	41: dst_hst_srv_rerrrt	

B. Parameter selection and alternative feature sets

Table III lists the various parameters of the feature selection algorithm and the classifier, as used during the experimental work.

TABLE III
FEATURE SELECTION AND CLASSIFIER PARAMETERS

Clustering	
Initialization	Random
Distance metric	Euclidean distance
Number of restarts (C_R)	100
Number of clusters (C_N)	5 – 10
Genetic Algorithm	
Individual	40-bit binary string
Population size	50
Number of generations	150
Minimum num. of features	15
Selected # of clusters (C_N)	7
Selection method	Stochastic uniform
Crossover type	Scattered
Mutation rate	Uniform at 1%
Stopping criteria	Maximum generations reached Avg change in obj. function $1e^{-4}$

The performance of the classifier, as used with the feature sets obtained from the proposed feature selection algorithm, was compared to the performance attained using several filter–based and wrapper–based feature sets. These feature sets are provided in initial four rows of table IV. Note that the wrapper–based feature sets were optimized for use with the k–means algorithm, in order to permit a fair comparison.

V. RESULTS

The feature selection algorithm (fig. 1) was used to produce two feature sets. The first set, referred to as 'GA–1' was obtained by using the genetic algorithm to optimize the single objective function, namely the normalized DB index (this corresponds to configuration 1, as set out in section III-A4). The second set, referred to as 'MOGA–1', was obtained by optimizing both the normalized DB index as well as the normalized candidate feature set cardinality (configuration 2). The two feature sets obtained with the two configurations are presented in the final rows of table IV.

TABLE IV
LIST OF FEATURE SETS (FEATURE NUMBERS AS PROVIDED IN TABLE II)

Name	Reference	Features
Filter–1 (7)	[20]	3-6,14,16,27,28,37,39
Filter–2 (11)	[20]	3,5,23,24
Wrapper–1 (FRM)	[10]	1,2,11,16,18,21,23,25–29,36,38,39
Wrapper–2 (SFM)	[10]	3,23,25,29,35
GA–1	Novel	2–4, 7, 8, 13–17, 21–29, 38–41
MOGA–1	Novel	2–5,7–9,14,15,18,21–24, 27,28,38,39,41

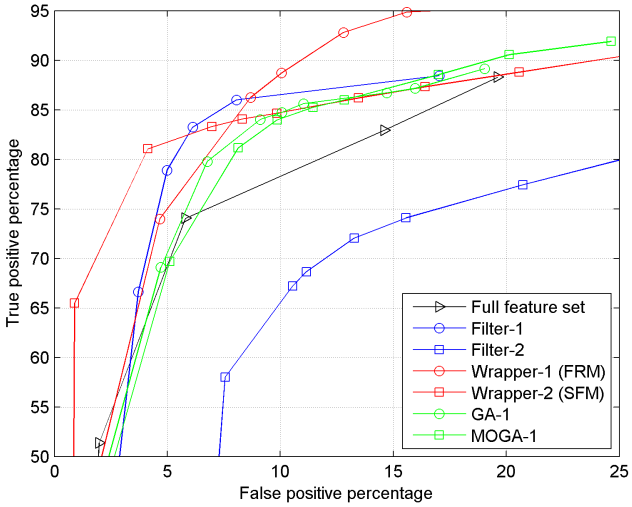

Fig. 2. Receiver operating curves for the proposed classifier, under various feature subsets.

The proposed feature sets, as well as the alternative feature sets of table IV, were used as input to the classifier in order to obtain receiver operating curves (refer to fig. 2). The figure reveals that the proposed feature selection algorithm produces feature sets that outperform the full feature set, when used to perform anomaly detection. In addition, the performance associated with the proposed feature sets approaches that of the existing filter–based and wrapper–based techniques. The proposed techniques are within 2% of the true positive rate of the best performing filter technique, for false positive rates exceeding 10%. Over the same range of the false positive rate, the algorithm has similar performance to the wrapper technique based on SFM, while the FRM–based wrapper technique outperforms the proposed technique. The proposed technique significantly outperforms the second filter technique.

The results of figure 2 are significant, as the proposed method does not rely on any knowledge of data labels, whereas the competing filter–based and wrapper–based methods, as represented in the figure, do require knowledge of data labels over a training subset of the data.

VI. Conclusion

In this paper, a feature selection algorithm was proposed for use in the context of network anomaly detection via clustering. The algorithm uses a genetic algorithm to optimize a cluster validity index over a search space consisting of feature subsets. The concept behind the proposed feature selection algorithm is that higher cluster validity indices, as individually computed after clustering of a dataset over a population of candidate feature subsets, are indicative of more relevant candidate feature subsets. The significant advantage of the proposed algorithm is that it is does not require any access to data labels or a training set in order to perform feature selection (i.e. it is unsupervised), as compared to existing feature selection techniques in the context of network intrusion detection. Results indicate that feature sets produced using the proposed technique correspond to classification performance that approaches that of existing filter–based and wrapper–based feature selection techniques.

References

[1] S. Kent, "On the trail of intrusions into information systems," *Spectrum, IEEE*, vol. 37, no. 12, pp. 52–56, 2000.

[2] D. K. Bhattacharyya and J. K. Kalita, *Network Anomaly Detection: A Machine Learning Perspective*. CRC Press, 2013.

[3] C. A. Catania and C. G. Garino, "Automatic network intrusion detection: Current techniques and open issues," *Computers & Electrical Engineering*, vol. 38, no. 5, pp. 1062–1072, 2012.

[4] M. Roesch *et al.*, "Snort: Lightweight intrusion detection for networks." in *LISA*, vol. 99, no. 1, 1999, pp. 229–238.

[5] V. Paxson, "Bro: a system for detecting network intruders in real-time," *Computer networks*, vol. 31, no. 23, pp. 2435–2463, 1999.

[6] P. Garcia-Teodoro, J. Diaz-Verdejo, G. Maciá-Fernández, and E. Vázquez, "Anomaly-based network intrusion detection: Techniques, systems and challenges," *computers & security*, vol. 28, no. 1, pp. 18–28, 2009.

[7] C.-F. Tsai, Y.-F. Hsu, C.-Y. Lin, and W.-Y. Lin, "Intrusion detection by machine learning: A review," *Expert Systems with Applications*, vol. 36, no. 10, pp. 11 994–12 000, 2009.

[8] L. Portnoy, E. Eskin, and S. J. Stolfo, "Intrusion detection with unlabeled data using clustering," in *Proceedings of The ACM Workshop on Data Mining Applied to Security*. ACM, 2001.

[9] J. J. Davis and A. J. Clark, "Data preprocessing for anomaly based network intrusion detection: A review," *Computers & Security*, vol. 30, no. 6, pp. 353–375, 2011.

[10] Y. Li, J. Xia, S. Zhang, J. Yan, X. Ai, and K. Dai, "An efficient intrusion detection system based on support vector machines and gradually feature removal method," *Expert Systems with Applications*, vol. 39, no. 1, pp. 424–430, 2012.

[11] A. Dastanpour and R. A. R. Mahmood, "Feature selection based on genetic algorithm and supportvector machine for intrusion detection system," in *The Second International Conference on Informatics Engineering & Information Science (ICIEIS2013)*. The Society of Digital Information and Wireless Communication, 2013, pp. 169–181.

[12] M. Tavallaee, "Nsl-kdd dataset," *http://www. iscx. ca/NSL-KDD*, 2012.

[13] I. Syarif, A. Prugel-Bennett, and G. Wills, "Unsupervised clustering approach for network anomaly detection," in *Networked Digital Technologies*. Springer, 2012, pp. 135–145.

[14] W. Lu and H. Tong, "Detecting network anomalies using cusum and em clustering," in *Advances in Computation and Intelligence*. Springer, 2009, pp. 297–308.

[15] M. H. Bhuyan, D. Bhattacharyya, and J. K. Kalita, "Nado: network anomaly detection using outlier approach," in *Proceedings of the 2011 International Conference on Communication, Computing & Security*. ACM, 2011, pp. 531–536.

[16] G. H. Orair, C. H. Teixeira, W. Meira Jr, Y. Wang, and S. Parthasarathy, "Distance-based outlier detection: consolidation and renewed bearing," *Proceedings of the VLDB Endowment*, vol. 3, no. 1-2, pp. 1469–1480, 2010.

[17] S. J. Stolfo. (1999, Oct.) Kdd datasets. [Online]. Available: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

[18] R. Kohavi and G. H. John, "Wrappers for feature subset selection," *Artificial intelligence*, vol. 97, no. 1, pp. 273–324, 1997.

[19] F. Amiri, M. Rezaei Yousefi, C. Lucas, A. Shakery, and N. Yazdani, "Mutual information-based feature selection for intrusion detection systems," *Journal of Network and Computer Applications*, vol. 34, no. 4, pp. 1184–1199, 2011.

[20] S. Zargari and D. Voorhis, "Feature selection in the corrected kdd-dataset," in *Emerging Intelligent Data and Web Technologies (EIDWT), 2012 Third International Conference on*. IEEE, 2012, pp. 174–180.

[21] S. Chebrolu, A. Abraham, and J. P. Thomas, "Feature deduction and ensemble design of intrusion detection systems," *Computers & Security*, vol. 24, no. 4, pp. 295–307, 2005.

[22] H. G. Kayacik, A. N. Zincir-Heywood, and M. I. Heywood, "Selecting features for intrusion detection: a feature relevance analysis on kdd 99 intrusion detection datasets," in *Proceedings of the third annual conference on privacy, security and trust*, 2005.

[23] A. A. Olusola, A. S. Oladele, and D. O. Abosede, "Analysis of kdd99 intrusion detection dataset for selection of relevance features," in *Proceedings of the World Congress on Engineering and Computer Science*, vol. 1. Citeseer, 2010, pp. 20–22.

[24] P. Tang, R.-a. Jiang, and M. Zhao, "Feature selection and design of intrusion detection system based on k-means and triangle area support vector machine," in *Future Networks, 2010. ICFN'10. Second International Conference on*. IEEE, 2010, pp. 144–148.

[25] M. Halkidi, Y. Batistakis, and M. Vazirgiannis, "On clustering validation techniques," *Journal of Intelligent Information Systems*, vol. 17, no. 2-3, pp. 107–145, 2001.

[26] G. Milligan and M. Cooper, "An examination of procedures for determining the number of clusters in a data set," *Psychometrika*, vol. 50, no. 2, pp. 159–179, 1985. [Online]. Available: http://dx.doi.org/10.1007/BF02294245

[27] D. L. Davies and D. W. Bouldin, "A cluster separation measure," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, no. 2, pp. 224–227, 1979.

[28] T. Caliński and J. Harabasz, "A dendrite method for cluster analysis," *Communications in Statistics-theory and Methods*, vol. 3, no. 1, pp. 1–27, 1974.

[29] Z. Ihsan, M. Y. Idris, and A. H. Abdullah, "Attribute normalization techniques and performance of intrusion classifiers: A comparative analysis." *Life Science Journal*, vol. 10, no. 4, 2013.

[30] J. MacQueen *et al.*, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, vol. 1, no. 14. Oakland, CA, USA., 1967, pp. 281–297.

[31] J. Handl and J. Knowles, "Feature subset selection in unsupervised learning via multiobjective optimization," *International Journal of Computational Intelligence Research*, vol. 2, no. 3, pp. 217–238, 2006.

[32] D. Beasley, R. Martin, and D. Bull, "An overview of genetic algorithms: Part 1. fundamentals," *University computing*, vol. 15, pp. 58–58, 1993.

Tyrone Naidoo received his bachelors degree in engineering from the University of Kwazulu-Natal in 2012. He is currently pursuing his masters degree in engineering at the same university under the CSIR Human Capital Development Programme in Information Security.

Andre M. McDonald received his masters in engineering from the University of Pretoria in 2010. He is currently a senior researcher at the Modelling and Digital Science unit of the Council for Scientific and Industrial Research.