EURASIP Journal on Wireless Communications and Networking

RESEARCH Open Access

Ensemble-based multi-filter feature selection method for DDoS detection in cloud computing

Opeyemi Osanaiye[1,2], Haibin Cai[3*], Kim-Kwang Raymond Choo[2], Ali Dehghantanha[4], Zheng Xu[5,6] and Mqhele Dlodlo[1]

Abstract

Widespread adoption of cloud computing has increased the attractiveness of such services to cybercriminals. Distributed denial of service (DDoS) attacks targeting the cloud's bandwidth, services and resources to render the cloud unavailable to both cloud providers, and users are a common form of attacks. In recent times, feature selection has been identified as a pre-processing phase in cloud DDoS attack defence which can potentially increase classification accuracy and reduce computational complexity by identifying important features from the original dataset during supervised learning. In this work, we propose an ensemble-based multi-filter feature selection method that combines the output of four filter methods to achieve an optimum selection. We then perform an extensive experimental evaluation of our proposed method using intrusion detection benchmark dataset, NSL-KDD and decision tree classifier. The findings show that our proposed method can effectively reduce the number of features from 41 to 13 and has a high detection rate and classification accuracy when compared to other classification techniques.

Keywords: Ensemble-based multi-filter feature selection method, Filter methods, Cloud DDoS, Intrusion detection system, Machining learning

1 Introduction

Cloud computing provides individual and organisational users the on-demand, scalable and reliable computing resources and can be deployed as a public, private, community or hybrid cloud. There are three main service models, namely software-as-a-service (SaaS), platform-as-a-service (PaaS) and infrastructure-as-a-service (IaaS) [1, 2]. Despite the benefits offered by the use of cloud computing, it could be exploited or targeted by cybercriminals, including state-sponsored actors (see [3]). This is not surprising, as popular consumer technologies such as wireless sensor networks have also been reportedly targeted [4, 5]. One common form of attacks targeting cloud computing is distributed denial of service (DDoS) attacks [6, 7], and we refer to interested reader to [8] for other cloud-related security and privacy issues. In its simplest form, a DDoS attacker seeks to compromise and take over hundreds to thousands vulnerable hosts, known as zombies, to facilitate or carry out a coordinated attack against a target. Such attacks have continued to increase both in size and sophistication, and extortion has been identified as one of the main motives behind such attacks [9].

Proposed DDoS defence techniques generally seek to classify packets as either legitimate or malicious and can be broadly categorised into signature-based or anomaly based. Signature-based techniques involve the use of attack signatures stored in a knowledge database to identify an attack, while anomaly based techniques use normal traffic behavioural pattern over a set period of time to determine whether subsequent patterns deviate from the expected behaviour. Signature-based detection is generally effective in detecting known attacks, while anomaly detection can potentially detect zero-day attacks. To overcome limitations associated with both approaches, hybrid solutions based on both techniques have been proposed in the literature [6].

* Correspondence: hbcai@sei.ecnu.edu.cn
[3]Shanghai Key Laboratory of Trustworthy Computing, East China Normal University, Shanghai, China
Full list of author information is available at the end of the article

Due to the increasing amount of data that needs to be processed [10–13], feature selection can be used in the pre-processing phase before classification in order to identify important features of a dataset, with the aims of improving prediction accuracy and reducing computational complexity. Existing defence methods that are capable of handling significant amount of data generally contain redundant or irrelevant features, which result in excessive training and classification time [14]. Feature selection methods have been used in a wide range of applications, such as statistical pattern recognition, machine learning and data mining for data reduction in order to achieve improved performance and detection of outliers. Current feature selection methods can be broadly categorised into filter, wrapper and embedded approaches. In filter methods, attributes are categorised according to the intrinsic information of the data and it is independent of the classification algorithm [15]. Features are then assessed and ranked according to their inherent properties using simple measurements such as distance, dependency and information [16]. Such methods are particularly efficient when dealing with large dataset, as compared to wrapper methods that provide a more precise result but are more time-consuming [17]. Wrapper and embedded methods require specific classification algorithm to determine the importance of a feature subset.

Recent studies have shown that combining feature selection methods can improve the performance of classifiers by identifying features that are weak as an individual but strong as a group [18], removing redundant features [17] and determining features that have a high correlation with the output class. Other methods have proposed a hybrid feature selection that combines both filter and wrapper. Filter feature selection represents a popular method that uses ranking and space search technique. Therefore, in this work, we present an ensemble-based multi-filter feature selection (EMFFS) method that combines the output of information gain (IG), gain ratio, chi-squared and ReliefF to select important features. The aim of this work is to significantly reduce the feature set while maintaining or improving the classification accuracy using a decision tree classifier. Intrusion detection benchmark dataset, NSL-KDD, consisting of 41 features is used to evaluate the efficiency of our proposed method in Waikato environment for knowledge analysis (Weka) [19].

The rest of the paper is organised as follows: related work is presented in Section 2 while the proposed EMFFS method is described in Section 3. In Section 4, the classification algorithm and benchmark dataset are presented. In Section 5, our experimental findings are discussed. Section 6 concludes the paper.

2 Related work

The performance of a classification problem depends on the relevance of the selected attributes with regard to its class. Feature selection methods have been applied in classification problems to select a reduced feature subset from the original set to achieve a faster and more accurate classification. Similar to many data mining and machine learning techniques, two key factors are involved in building an optimum classifier: feature and model selection [20]. Selecting the right feature can be quite a challenging task, and several methods have been proposed to solve this and discard redundant, irrelevant and noisy features.

Wang and Gombault [21] proposed a filter selection method using IG and chi-squared to extract nine most important features in the network traffic. Bayesian network and C 4.5 (a decision tree classifier) were used to detect DDoS attack in the network. Results obtained show that the detection accuracy remains the same while the overall efficiency improved. Bolon-Canedo et al. [18] combined discretizers, filters and classifiers to improve the classification performance by significantly reducing the feature set. This was applied to both binary and multi-class classification problems using KDD Cup '99 benchmark dataset. A supervised inductive learning approach, group method for data handling (GMDH), was proposed in [22] using monolithic and ensemble-based techniques. Filter feature selection methods using IG, gain ratio and GMDH were used to rank features during the pre-processing phase. Lin et al. [23] proposed an anomaly intrusion detection that detects new attacks using support vector machine (SVM), decision tree (DT) and simulated annealing (SA). The best features were selected from the KDD '99 dataset using SVM and SA to improve the classification accuracy of DT and SVM, to detect new attacks. Li et al. [24] proposed a gradual feature removal method that process dataset prior to combining cluster method, ant colony algorithm and SVM to classify network traffic as either normal or anomaly. Sindhu et al. [25] proposed a wrapper method for feature selection to remove irrelevant instances from a feature set to achieve higher detection accuracy using neuro tree. A feature selection approach was proposed in [26] using Bayesian network, and NSL-KDD dataset was used to evaluate the selected features. Findings indicated that these features decreased the attack detection time and improved the classification accuracy as well as the true positive rates. Bhattacharya et al. [27] proposed a multi-measure multi-weight ranking approach that identifies important network features by combining wrapper, filter and clustering methods to assign multiple weights to each feature.

Rough set feature selection approach has proven to be an efficient mathematical tool based on upper and lower approximation. It presents equal classification capability with minimal subset. Olusola et al. [28] proposed a rough set-based feature selection method that selects important features from an input data using KDD '99 dataset. Sengupta et al. [29] designed an online intrusion detection system (IDS) using both rough set theory and Q-learning algorithm to achieve a maximum classification algorithm that classifies data as either normal or anomaly using NSL-KDD network traffic data. A fast attribute reduction algorithm based on rough set theory was proposed in [30]. The algorithm identifies important features and discards independent and redundant attributes to achieve an effective classification performance.

A review of the literature suggests that there are three general trends in feature selection, irrespective of the method used, namely (1) methods proposed search and identify correlated features in the dataset in order to remove the redundant features,(2) methods identify unique features that contain important information about different output classes in the data and discard ones with little or no information and (3) some features have been identified to be strong as a group but weak individually. In filter feature selection method, features are ranked independently according to their strength in predicting the output class. Unlike previously proposed methods, this work presents an algorithm that supports data mining and security defence for cloud DDoS attacks using minimal feature set. Filter feature selection methods present different ranking algorithms; therefore, we propose an EMFFS method that combines the output of IG, gain ratio, chi-squared and ReliefF to find common features in the one-third split of the ranked features using NSL-KDD benchmark dataset in the Weka. We, therefore, reduce the features from 41 to 13 and use J.4.8, a version of C4.5 decision tree classification algorithm to classify data as either normal or anomaly.

3 EMFFS method

The filter feature selection method is a pre-processing phase towards selecting important features from a dataset and is independent of the classification algorithm. Filter methods rely on statistical intrinsic test over an original training dataset and use a feature ranking scheme as the main criteria for feature selection by ordering. Features are scored, and a pre-determined threshold is used to remove features below the threshold. Due to its simplicity, it has been widely used for practical applications, including cloud computing, involving a huge amount of data. In this section, we describe our proposed ensemble-based multi-filter feature selection method that combines the output of four filter selection methods—IG, gain ratio, chi-squared and ReliefF—to harness their combined strength to select 13 common features among them.

A. Information gain

One of the filter feature selection methods used in determining relevant attributes from a set of features is IG. IG works by reducing the uncertainty associated with identifying the class attribute when the value of the feature is unknown [31]. It is based on information theory which is used in ranking and selecting top features to reduce the feature size before the start of the learning process. The entropy value of the distribution is measured to determine the uncertainty of each feature prior to ranking, according to their relevance in determining different classes [32]. The uncertainty is determined by the entropy of the distribution, sample entropy or estimated model entropy of the dataset. The entropy of variable X [33] can be defined as:

$$H(X) = -\sum_{i} P(x_i)\log_2(P(x_i)) \tag{1}$$

Let $P(x_i)$ denotes the value of prior probabilities of X. The entropy of X after observing value of another variable Y is defined as:

$$H(X/Y) = -\sum_{j} P\left(y_j\right)\sum_{i} P\left(x_i|y_j\right)\log_2\left(P\left(x_i|y_j\right)\right) \tag{2}$$

In Eq. 2, $P(x_i|y_j)$ is the posterior probability of X given Y. The information gain is defined as the amount by which the entropy of X decreases to reflect an additional information about X provided by Y and is defined as:

$$IG(X/Y) = H(X) – H(X|Y) \tag{3}$$

Based on this measure, it is clear that features Y and X are more correlated than features Y and Z, if $IG(X/Y) > IG(Z/Y)$. The feature ranking can, therefore, be calculated using Eq. 3. This ranking will be used to select the most important features.

B. Gain ratio

The gain ratio was introduced to improve the bias of IG towards features with large diversity value [22]. When data are evenly spread, gain ratio exhibits a high value while it gives a small value when all data belongs to only one branch of the attribute. It uses both the number and size of branches to determine an attribute and corrects IG by considering intrinsic information [34]. The intrinsic information of a given feature can be determined by the entropy distribution of the instance value. Gain ratio of a given feature x and a feature value y can be calculated [34] using Eqs. 4 and 5.

$$\text{Gain Ratio }(y,\ x) = \frac{\text{Information Gain}(y,\ x)}{\text{Intrinsic Value}(x)}, \qquad (4)$$

where

$$\text{Intrinsic Value }\ (x) = \ -\sum \frac{|S_i|}{|S|} * \text{Log}_2 \frac{|S_i|}{S} \qquad (5)$$

Note that $|S|$ is the number of possible values feature x can take, while $|S_i|$ is the number of actual values of feature x. In our work, we selected 14 features, representing one-third split of the ranked features using NSL-KDD benchmark dataset. These 14 features represent the highest ranked feature using gain ratio.

C. Chi-squared

The chi-squared (χ^2) statistic is used to test the independence of two variables by computing a score to measure the extent of independence of these two variables. In feature selection, χ^2 measures the independence of features with respect to the class. The initial assumption of χ^2 is that the feature and the class are independent before computing a score [35]. A score with a large value indicates the existence of a high-dependent relationship. Chi-squared [36] can be defined as:

$$\chi^2(r,\ c_i) = \frac{N[P(r,\ c_i)P(\bar{r},\bar{c}_i) - P(r,\bar{c}_i)P(\bar{r},\ c_i)]^2}{P(r)P(\bar{r})P(c_i)P(\bar{c}_i)}, \qquad (6)$$

where N denotes the entire dataset, r indicates the presence of the feature ($\bar{r}$ its absence) and c_i refers to the class. $P(r,\ c_i)$ is the probability that feature r occurs in class c_i, and $P(\bar{r},\ c_i)$ is the probability that the feature r does not occur in class c_i. Also, $P(r,\bar{c}_i)$ and $P(\bar{r},\bar{c}_i)$ are the probabilities that the features do or do not occur in a class that is not labelled c_i and so on. $P(r)$ is the probability that the feature appears in the dataset while $P(\bar{r})$ is the probability that the feature does not appear in the dataset. $P(c_i)$ and $P(\bar{c}_i)$ are the probabilities that a dataset is labelled to class c_i or not.

D. ReliefF

ReliefF feature selection method uses continuous sampling to evaluate the worth of a feature to distinguish between the nearest hit and nearest miss (nearest neighbour from the same class and from a different class) [37]. The attribute evaluator is used to append weight to each feature according to its ability to distinguish the different classes. A user-defined threshold is determined, and weight of features that exceeds this threshold is selected as important features [35]. ReliefF evolved from the original Relief algorithm [38] and was developed to improve its limitations. Among the key attributes of ReliefF are its ability to deal with the multi-class problem and its robustness and capability to deal with noisy and incomplete data. A key advantage of ReliefF over other filter methods is that it has a low bias and can be applied in all situations.

3.1 EMFFS execution process

Our proposed EMFFS method uses the output of the one-third split of ranked features of the filter methods described above. EMFFS is a pre-processing phase prior to learning, where individual filter methods are used for the initial selection process. IG, gain-ratio, chi-square and ReliefF filter methods are used to rank the feature set of the original dataset to create a mutually exclusive subset before selecting one-third split of the ranked features (i.e. 14 features). These features are considered as the most important feature with respect to each filter method.

The resulting output of the EMFFS is determined by combining the output of each filter method and using a simple majority vote to determine the final selected feature. A threshold is determined to identify the frequently occurring features among the four filter methods and set to 3 (i.e. $T = 3$). After combining all the selected feature sets, a counter is used to determine common features with respect to the threshold set. Features that meet the threshold criteria are selected and used as the final feature set for classification. Figure 1 shows the proposed EMFFS method.

The EMFFS method is constructed through the algorithms presented below.

```
Algorithm 3.1.1 (Filter feature ranking methods)
  Step 1: Let $X_i$ be the feature set in the NSL-KDD
          dataset, where $X_i = \{X_1,\ X_2,\ X_3\ \dots\ \dots\ \dots\ \dots\ ,\ \ X_{41}\ \}$ and $C_i$ represents the class (i.e.
          normal or anomaly), where $C_i = \{\ C_1, C_2\}$.
  Step 2: For each filter method, rank and sort the
          features $X_i$ according to its importance in
          determining the output class $C_i$.
  Step 3: Select one-third split of each filter selection
          method's output $X_i^{'}$.

Algorithm 3.1.2 (Combine output features)
  Step 1: Combine selected output features $X_i^{'}$ of each
          filter method.
  Step 2: Determine the feature count threshold $T$.
  Step 3: Compute the feature occurrence rate among
          the filter methods.

Algorithm 3.1.3 (Ensemble selection)
  Step 1: Choose intercepts of common features from 3.1.2
  Step 2: If the feature count is less than the threshold,
          drop the feature otherwise select the feature.
  Step 3: Repeat step 2 for all the features in the one-
          third split subset.
```

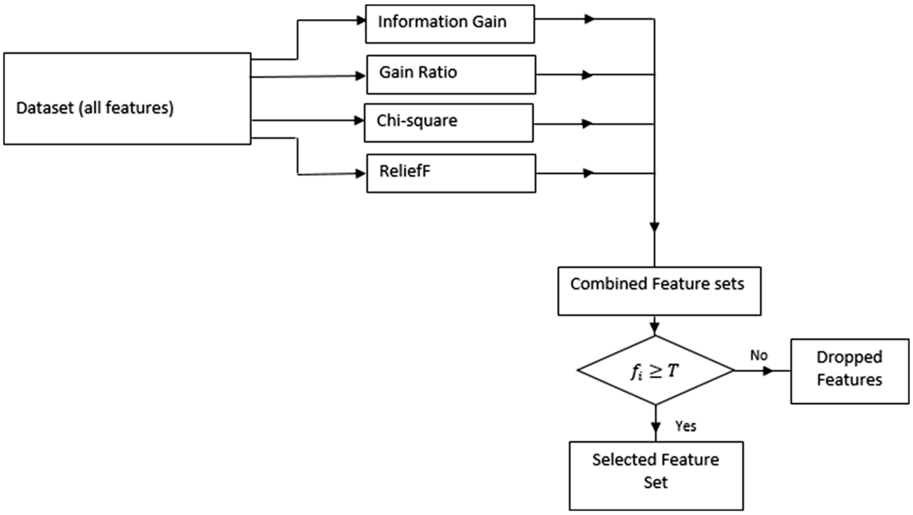

Fig. 1 Proposed Ensemble-based Multi-Filter Feature Selection (EMFFS) Method

4 Classification algorithm and dataset

Decision tree classification algorithm is a popular data mining classifier for prediction due to the ease of understating and the interaction between variables. It is based on a greedy algorithm that uses a divide-and-conquer strategy to recursively construct a decision tree [39]. The tree is made up of a root node, internal nodes, branches and leaves, which represents a rule used in categorising data according to its attributes. Decision tree uses supervised dataset with root node being the first attribute with the test condition to split each input towards individual internal node, in line with the characteristics of the data record [38]. The node with the highest information gain is the root node, and the preceding node with the next highest information gain is selected as the test for the next node. This process continues until all attributes have been compared or when all the samples belong to the same class with no remaining attribute to which the samples can be further partitioned [40].

A branch connects two nodes together and can also connect a node and a leaf. Each node is made up of branches labelled as the possible value of attributes in the parent node [23]. The leaves are labelled as the decision value of classification.

Consider a case selected at random from a set S of cases which belongs to class $C_{i.}$ The probability that an arbitrary sample belongs to class C_i can be determined as follows [40]:

$$P_i = \frac{\text{freq}(C_\text{i},S)}{|S|}, \tag{7}$$

where $|S|$ is the number of samples in the set S. Therefore, the information it convey can be represented by $-\log_2 P_i$

Table 1 NSL-KDD dataset features

Number	Data features	Number	Data features	Number	Data features	Number	Data features
1	Duration	12	Logged_in	23	Count	34	Dst_host_same_srv_rate
2	Protocol_type	13	Num_compromised	24	Srv_count	35	Dst_host_diff_srv_rate
3	Service	14	Root_shell	25	Serror_rate	36	Dst_host_same_src_port_rate
4	Flag	15	Su_attempted	26	Srv_serror_rate	37	Dst_host_srv_diff_host_rate
5	Src_bytes	16	Num_root	27	Rerror_rate	38	Dst_host_serror_rate
6	Dst_bytes	17	Num_file_creations	28	Srv_rerror_rate	39	Dst_host_srv_serror_rate
7	Land	18	Num_shells	29	Same_srv_rate	40	Dst_host_rerror_rate
8	Wrong_fragment	19	Num_access_files	30	Diff_srv_rate	41	Dst_host_srv_rerror_rate
9	Urgent	20	Num_outbound_cmds	31	Srv_diff_host_rate		
10	Hot	21	Is_host_login	32	Dst_host_count		
11	Num_failed_logins	22	Is_guest_login	33	Dst_host_srv_count		

bits. Now, suppose the probability distribution is given as $P = \{P_1, P_2, P_3 \ldots \ldots \ldots \ldots, \ P_n\}$, therefore, the information carried by the distribution, that is entropy of P, can be expressed as:

$$\text{Info } (P) = \sum_{i=1}^{n} -P_i \ \log_2 P_i \tag{8}$$

Partitioning a set of K samples, based on the value of a non-categorical attribute X, into sets $K_1, K_2, K_3 \ldots \ldots \ldots \ldots, \ K_m$, the information required to determine the class of an element of K is the weighted average of the information needed to identify the class of an element K_i. The weighted average of Info (K_i) can be determined by:

$$\text{Info } (X, K) = \sum_{i=1}^{m} \frac{|K_i|}{K} \times \text{ Info}(K_i) \tag{9}$$

The information gain, Gain (X, K), can therefore be calculated as follows:

$$\text{Gain } (X, K) = \text{Info } (K) - \text{Info } (X, K) \tag{10}$$

Equation 10 represents the difference between the information needed to identify an element of K and the information needed to identify an element of K after the value of attribute X has been determined. Therefore, this is the information gain due to attribute X.

There are different algorithms for implementing decision tree; C5.0 and its earlier version C4.5 have been described in [41]; however, for our work, we will use J48, a version of C4.5 as our classifier.

4.1 Benchmark datasets

NSL-KDD dataset, an improved version of KDD CUP '99 widely deployed in the literature [26, 29, 42] for intrusion detection, was used to validate our proposed algorithm. NSL-KDD is a labelled benchmark dataset from KDD CUP '99 to improve its flaws. Researchers have identified several issues associated with the use of KDD CUP '99, such as existence of large redundant records (which may result in learning algorithm being biased towards frequently occurring records) and its high complexity [43]. NSL-KDD is used for evaluating network intrusion systems and is made up of selected records from the initial KDD CUP '99. This presents a reduced dataset size that makes the evaluation of different research works consistent and validation of learning algorithm complete, easy and affordable. NSL-KDD is made up of 41 features and labelled as either attack or normal (see Table 1). These features are categorised into four groups, namely basic, content, time-based traffic and connection-based traffic [27]. NSL-KDD comprises both training and testing datasets. The former is made up of 21 attack types while an additional 17 novel attack types are used for the test set [26]. The attacks are grouped into four categories: DoS, Probe, U2R and R2L. While the distribution of the training dataset consists of 67,343 normal (53.46 %), 45,927 DoS (36.46 %), 11,656 Probe (9.25 %), 995 R2L (0.79 %) and 52 (0.04 %) U2R, the testing dataset on the other hand contains 9711 normal (43.08 %), 7456 DoS (33.08 %), 2421 probe (10.74 %), 2756 R2L (12.22 %) and 200 U2R (0.89 %).

From the attack distribution, DoS constitutes around 78.3 % of the total attack. Therefore, in this work, we used 20 % of the records in NSL-KDD train+ as our DoS training set that has been labelled as either attack or normal. We apply 10-fold cross-validation for both training and testing purpose. Table 1 describes the NSL-KDD feature dataset.

Table 2 Feature selection using filter methods

Filter method	Feature selected
Info gain	5,3,6,4,30,29,33,34,35,38,12,39,25,23
Gain ratio	12,26,4,25,39,6,30,38,5,29,3,37,34,33
Chi-squared	5,3,6,4,29,30,33,34,35,12,23,38,25,39
ReliefF	3,29,4,32,38,33,39,12,36,23,26,34,40,31

Table 3 Ensemble-based multi-filter feature selection (EMFFS) method

Filter method	Feature selected
EMFFS	3,4,29,33,34,12,39,5,30,38,25,23,6

5 Experimental results

In this section, we deployed our proposed EMFFS method to pre-process the dataset to select the most important features for decision tree classification algorithm that classifies data as either attack or normal in cloud computing. Our analysis were carried out using Weka [44] that contains a collection of machine learning algorithms for data mining tasks. The parameters for classification in the experiments were set to the default values in Weka.

Table 4 Performance measure

Filter method	No. of features	Accuracy (%)	Detection rate (%)	False alarm rate (%)	Time (s)
Full set	41	99.56	99.49	0.38	2.75
Info gain	14	99.66	99.74	0.41	0.83
Gain ratio	14	99.60	99.68	0.47	1.12
Chi-squared	14	99.66	99.74	0.41	0.92
ReliefF	14	99.08	99.02	0.87	0.93
EMFFS	13	99.67	99.76	0.42	0.78

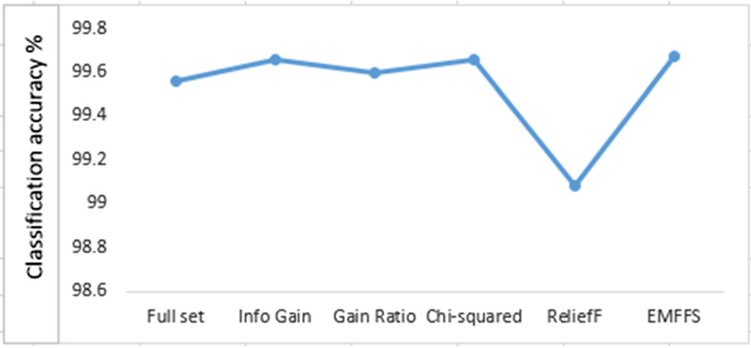

Fig. 2 Classification accuracy for filter methods

We used NSL-KDD dataset to evaluate the performance of our EMFFS method and decision tree classifier using 10-fold cross-validation. In the 10-fold cross-validation, data was divided into 10-fold of equal sizes before performing 10 iterations of training and validation. Within each iteration, a different fold of the data was used for validation while the remaining ninefold are used for learning. All experiments were performed on a 64-bit Windows 8.1 operating system with 6 GB of RAM and Intel core i5-4210U CPU.

5.1 Pre-processing dataset

During the pre-processing phase, feature selection was performed to determine the most important features of NLS-KDD dataset, by ranking them, using different filter methods. Fourteen most important features of the filter methods were determined by presenting one-third split of the ranked features (see Table 2).

After applying algorithm 3.1.2 to the output of each of the four filter selection method, we searched for feature intercept and set the minimum threshold to 3. From Table 2, it is observed that, even though each filter uses different ranking techniques, some features are common across different filter methods. Using simple majority vote, features 4, 29, 34, 12, 39, 3, 5, 6, 30, 33, 38, 25, and 23 (indicated in bold) appeared across more than three filter methods; this shows the level of importance these features are to the output class (see Table 3).

Table 3 shows the 13 selected features out of the one-third split of the most important features of NSL-KDD dataset using EMFFS method. This were used as the input features for training the decision tree classification algorithm, J48, in Weka.

5.2 Performance measures

The performance of a classifier could be determined by using different metrics. Determining the accuracy usually involves the measure of true positive (TP), true negative (TN), false positive (FP) and false negative (FN). TP is the amount of attack classified correctly while TN is the percentage of normal test sample classified correctly. FP is the amount of attack detected when it is indeed normal (false alarm), and FN is the misclassification of a test sample as normal when it is actually an attack.

Recently proposed mitigation strategies require high detection rate and low false alarm; therefore, in this work, we compared the accuracy, detection rate and

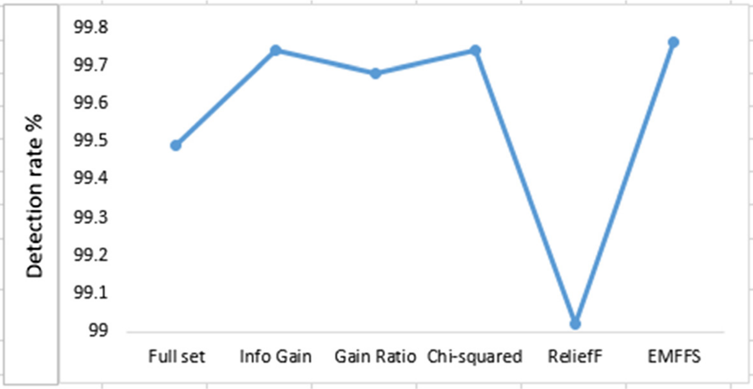

Fig. 3 Detection rate for filter methods

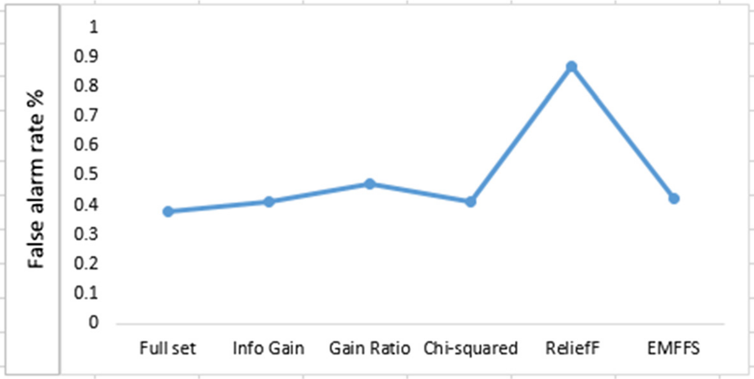

Fig. 4 False alarm rate for filter methods

false alarm rate of our proposed EMFFS method with each filter method and the full dataset feature using J48 classification algorithm. Furthermore, we compared the time required to build the classification model, which is the duration of the classifier's learning process after applying each feature selection method.

Table 4 presents the results of the performance measure of the J48 classifier using the full dataset with 41 features, one-third split of filter methods with 14 features and our proposed EMFFS method with 13 features.

5.2.1 Classification accuracy

Classification accuracy is the percentage of correctly defined data from the total set represented by the situation of TP and TN. The accuracy of the classifier can be determined by $cy = \frac{TP+TN}{TP+TN+FP+FN} \times 100\,\%$. Figure 2 shows the classification accuracy across different filter feature selection methods and EMFFS method. Our proposed method presents a slight increase in classification accuracy by 0.01 %.

5.2.2 Detection rate

Detection rate can be determined based on the confusion matrix. It is calculated as $\text{detection rate} = \frac{TP}{TP+FN} \times 100\,\%$. Figure 3 shows the performance of EMFFS method in comparison to other filter feature selection methods. The findings demonstrated that our method, with 13 selected features, has a slight increase in detection rate by 0.02 % when compared with the best filter feature selection method.

5.2.3 False alarm rate

False alarm is the amount of normal data that has been falsely classified as an attack, this can be determined by $\text{False alarm rate} = \frac{FP}{FP+TN} \times 100\,\%$. Figure 4 shows the false alarm rate of the full feature set and different filter feature selection methods. ReliefF produces the highest false alarm rate while the full feature set has the lowest rate with 0.38 %. Our method presents a false alarm rate of 0.42 %.

5.2.4 Time to build model

Figure 5 presents the time to build model across different filter selection methods and the full feature set. Our proposed method presents the best time with 0.78 s, when compared with other filter selection methods. The full feature set presents the worst learning

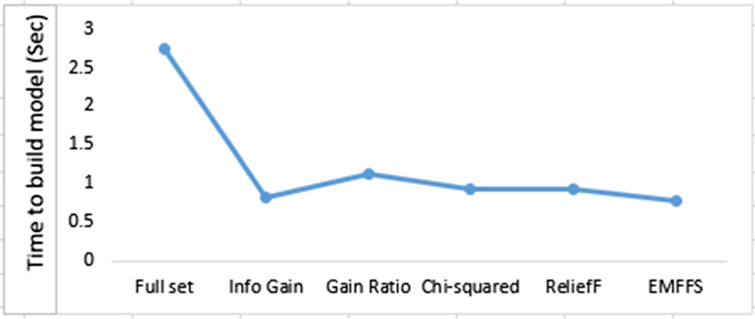

Fig. 5 Time to build model for filter methods

Table 5 Performance comparison with other feature selection approaches

Approach	Classifier	No. of features	Accuracy (%)	Time to build model (s)
CFS [44]	C4.5	NA	99.13	NA
CFS, CONS and INTERACT [20]	HNB_PKI_INT	7	93.72	NA
Gradual feature removal [14]	Cluster methods, ant colony algorithm and SVM	19	98.62	NA
CSE and CFS [45]	GA	32	78	NA
Linear correlation-based [42]	C.45	17	99.1	12.02
Our method (EMFFS)	J48	13	99.67	0.78

NA not available

time with 2.75 s. This is due to the number of features the classifier have to process.

5.3 Discussion

The need for effective real-time classification of DDoS attack in cloud computing increases the complexity of detection techniques. Filter methods for feature selection have proven to be crucial when designing a lightweight detection system, which involves identifying important features. In our proposed EMFFS method, we selected 13 features out of available 41 features by first presenting the output of one-third split using four filter methods. We determined a threshold and used a counter to select important features by simple majority voting. We compared our EMFFS method with other filter methods with 14 features and the full set consisting of 41 features using J48 decision tree classifier. Our method with 13 features presents an improvement in classification accuracy and detection rate. This implies that the original dataset contains some level of redundant feature that has little or no contribution towards identifying a particular class. For the time taken to build the model, our proposed method presents the best time when compared with individual filter selection methods and the full feature set. This makes our ensemble-based multi-filter feature selection method efficient with less complexity.

We then compared the performance of our proposed method, EMFFS, with methods proposed in the literature, by considering numbers of feature selected, classification accuracy and time to build model (see Table 5). We observed that using 13 most important features with decision tree classifier, our method provides the best classification accuracy and a better learning time when compared with other schemes presented in Table 5.

6 Conclusions

One of the notable challenges faced by current network intrusion systems in cloud computing is the handling of massive internet traffic during DDoS attacks. Feature selection methods have been used to pre-process dataset prior to attack classification in cloud computing. This work presented an ensemble-based multi-filter feature selection method that combines the output of one-third split of ranked important features of information gain, gain ratio, chi-squared and ReliefF. The resulting output of the EMFFS is determined by combining the output of each filter method. We used a set threshold to determine the final features using a simple majority vote. Performance evaluation with NSL-KDD dataset demonstrated that EMFFS method, with 13 features, achieves better performance than individual filter feature selection methods using J48 classifier and other proposed feature selection methods in the literature.

In the future, we seek to extend our work to include other classification algorithms and evaluate using other publicly available labelled datasets.

Competing interests
The authors declare that they have no competing interests.

Acknowledgements
The authors would like to thank the three anonymous reviewers for providing the constructive and generous feedback. Despite their invaluable assistance, any errors remaining in this paper are solely attributed to the authors. Opeyemi Osanaiye and Ali Dehghantanha acknowledged the financial support of The University of Cape Town, South Africa, and The University of Salford, UK, respectively. Haibin Cai is supported by the Natural Science Foundation of China under Grant No. 91118008 and Shanghai Knowledge Service Platform for Trustworthy Internet of Things Grant No. ZF1213.

Author details
[1]Department of Electrical Engineering, University of Cape Town, Rondebosch, South Africa. [2]Information Assurance Research Group, University of South Australia, South Australia 5095, Australia. [3]Shanghai Key Laboratory of Trustworthy Computing, East China Normal University, Shanghai, China. [4]School of Computing, Engineering and Technology, University of Salford, Manchester, UK. [5]Tsinghua University, Beijing, China. [6]Third Research Institute of the Ministry of Public Security, Shanghai, China.

Received: 11 February 2016 Accepted: 24 April 2016
Published online: 10 May 2016

References

1. O Osanaiye, M Dlodlo, Proceedings of 16th International Conference on Computer as a Tool (EUROCON), in *TCP/IP header classification for detecting spoofed DDoS attack in Cloud environment* (IEEE, Salamanca, 2015), pp. 1–6
2. O Osanaiye, Proceedings of 18th International Conference on Intelligence in Next Generation Networks (ICIN), in *Short Paper: IP spoofing detection for preventing DDoS attack in Cloud Computing* (IEEE, Paris, 2015), pp. 139–141
3. C Esposito, M Ficco, F Palmieri, A Castiglione, Interconnecting federated clouds by using publish-subscribe service. Cluster comput **16**(4), 887–903 (2013)
4. S Shamshirband, NB Anuar, ML Kiah, VA Rohani, D Petković, S Misra, AN Khan, Co-FAIS: cooperative fuzzy artificial immune system for detecting intrusion in wireless sensor networks. J Netw Comput Appl **42**, 102–117 (2014)
5. S Shamshirband, A Patel, NB Anuar, MLM Kiah, A Abraham, Cooperative game theoretic approach using fuzzy Q-learning for detecting and preventing intrusions in wireless sensor networks. Eng Appl Artif Intell **32**, 228–241 (2014)

6. O Osanaiye, K-KR Choo, M Dlodlo, Distributed denial of service (DDoS) resilience in cloud: review and conceptual cloud DDoS mitigation framework. J. Netw. Comput. Appl **67**, 147–65 (2016)
7. M Ficco, M Rak, Stealthy denial of service strategy in cloud computing. IEEE Trans Cloud Comput **3**(1), 80–94 (2015)
8. AN Khan, MM Kiah, SA Madani, M Ali, S Shamshirband, Incremental proxy re-encryption scheme for mobile cloud computing environment. J Supercomput **68**(2), 624–651 (2014)
9. L Krämer, J Krupp, D Makita, T Nishizoe, T Koide, K Yoshioka, C Rossow, *Proceedings of 18th International Symposium on Research in Attacks Intrusion and Defenses (RAID). AmpPot: monitoring and defending against amplification DDoS attacks* (Springer, Kyoto, 2015), pp. 615–636
10. L Zhao, L Chen, R Ranjan, K-K R Choo and J He, Geographical information system parallelization for spatial big data processing: a review. Cluster Comput. (Springer, 2015 in press)
11. D Quick, K-KR Choo, Impacts of increasing volume of digital forensic data: a survey and future research challenges. Digit Investig **11**(4), 273–294 (2014)
12. Z Xu, Y Liu, N Yen, L Mei, X Luo, X Wei, C Hu, Crowdsourcing based description of urban emergency events using social media big data. IEEE Trans Cloud Comput (2016) DOI: 10.1109/TCC.2016.2517638.
13. Z Xu, H Zhang, V Sugumaran, K-KR Choo, L Mei, Y Zhu, Participatory sensing-based semantic and spatial analysis of urban emergency events using mobile social media. EURASIP J Wirel Commun Netw **1**, 1–9 (2016)
14. J Peng, K-K R Choo, H Ashman, Bit-level n-gram based forensic authorship analysis on social media: Identifying individuals from linguistic profiles. J Netw Comput Appl. (Elsevier, 2016 in press)
15. P Bermejo, L de la Ossa, J Gámez, J Puerta, Fast wrapper feature subset selection in high-dimensional datasets by means of filter re-ranking. Knowl-Based Syst **25**(1), 35–44 (2012)
16. Y Chen, Y Li, X Cheng, L Guo, Proceedings of the 2nd SKLOIS Conference Information Security and Cryptology (INSCRYPT), in *Survey and taxonomy of feature selection algorithms in intrusion detection system* (Springer, Beijing, 2006), pp. 153–167
17. W Wang, Y He, J Liu, S Gombault, Constructing important features from massive network traffic for lightweight intrusion detection. IET Inform Secur **9**(6), 374–379 (2015)
18. V Bolon-Canedo, N Sanchez-Marono, A Alonso-Betanzos, Feature selection and classification in multiple class datasets: an application to KDD Cup 99 dataset. Expert Syst Appl **38**(5), 5947–5957 (2011)
19. Data mining software in Java. http://www.cs.waikato.ac.nz/ml/weka/(Last accessed 7 February 2016)
20. L Koc, T Mazzuchi, S Sarkani, A network intrusion detection system based on a Hidden Naïve Bayes multiclass classifier. Expert Syst Appl **39**(18), 13492–13500 (2012)
21. W Wang, S Gombault, W Wang, S Gombault, Proceedings of the 3rd International conference on Risks and Security of Internet and Systems (CRiSIS'08), in *Efficient detection of DDoS attacks with important attributes* (IEEE, Tozeur, 2008), pp. 61–67
22. Z Baig, S Sait, A Shaheen, GMDH-based networks for intelligent intrusion detection. Eng Appl Artif Intel **26**(7), 1731–1740 (2013)
23. S Lin, K Ying, C Lee, Z Lee, An intelligent algorithm with feature selection and decision rules applied to anomaly intrusion detection. Appl Soft Comput **12**(10), 3285–3290 (2012)
24. Y Li, J Xia, S Zhang, J Yan, X Ai, K Dai, An efficient intrusion detection system based on support vector machines and gradually feature removal method. Expert Syst Appl **39**(1), 424–430 (2012)
25. S Sindhu, S Geetha, A Kannan, Decision tree based light weight intrusion detection using a wrapper approach. Expert Syst Appl **39**(1), 129–141 (2012)
26. F Zhang, D Wang, Proceedings of the 8th International Conference on Networking, Architecture and Storage (NAS), in *An effective feature selection approach for network intrusion detection* (IEEE, Xi'an, 2013), pp. 307–311
27. S Bhattacharya, S Selvakumar, Multi-measure multi-weight ranking approach for the identification of the network features for the detection of DoS and Probe attacks. Compt. J. 1-21 (2015)
28. A Olusola, A Oladele, D Abosede, in *Proceedings of the World Congress on Engineering and Computer Science*. Analysis of KDD'99 intrusion detection dataset for selection of relevance features (San Francisco, USA, 2010), pp.1-7. http://www.iaeng.org/publication/WCECS2010/WCECS2010_pp162-168.pdf
29. N Sengupta, J Sen, J Sil, M Saha, Designing of on line intrusion detection system using rough set theory and Q-learning algorithm. Neurocomputing **111**, 161–168 (2013)
30. G Geng, N Li, S Gong, The Proceedings of International Conference on Industrial Control and Electronics Engineering (ICICEE), in *Feature Selection Method for Network Intrusion Based on Fast Attribute Reduction of Rough Set* (IEEE, Xi'an, 2012), pp. 530–534
31. B Agarwal, N Mittal, *Proceedings of 14th International Conference on Computational Linguistics and Intelligent Text Processing (CICLing). Optimal feature selection for sentiment analysis* (Springer, Samos, Greece, 2013), pp. 13–24
32. A Tesfahun, D Bhaskari, Proceedings of the International Conference on Cloud & Ubiquitous Computing & Emerging Technologies (CUBE), in *Intrusion Detection using Random Forests Classifier with SMOTE and Feature Reduction* (IEEE, Pune, 2013), pp. 127–132
33. L Yu, H Liu, *Proceedings of the Twentieth International Conference on Machine Learning (ICML-2003). Feature selection for high-dimensional data: A fast correlation-based filter solution* (Springer, Washington DC, 2003), pp. 856–863
34. H Ibrahim, S Badr, M Shaheen, Adaptive layered approach using machine learning techniques with gain ratio for intrusion detection systems. Int J Comput Appl **56**(7), 10–16 (2012)
35. L Devi, P Subathra, P Kumar, *Proceedings of the Fifth International Conference on Fuzzy and Neuro Computing (FANCCO-2015). Tweet sentiment classification using an ensemble of machine learning supervised classifiers employing statistical feature selection methods* (Springer, Hyderabad, 2015), pp. 1–13
36. N Nissim, R Moskovitch, L Rokach, Y Elovici, Detecting unknown computer worm activity via support vector machines and active learning. Pattern Anal Appl **15**(4), 459–475 (2012)
37. M Moradkhani, A Amiri, M Javaherian, H Safari, A hybrid algorithm for feature subset selection in high-dimensional datasets using FICA and IWSSr algorithm. Appl Soft Comput **35**, 123–135 (2015)
38. N Sánchez-Maroño, A Alonso-Betanzos, M Tombilla-Sanromán, *Proceedings of the Eighth International Conference on Intelligent Data Engineering and Automated Learning (IDEAL 2007). Filter methods for feature selection—a comparative study* (Springer, Birmingham, 2007), pp. 178–187
39. J Gehrke, V Ganti, R Ramakrishnan, W Loh, Proceedings of the International Conference on Management of Data, in *BOAT—optimistic decision tree construction* (ACM SIGMOD, Philadelphia, 1999), pp. 169–180
40. C Xiang, P Yong, L Meng, Design of multiple-level hybrid classifier for intrusion detection system using Bayesian clustering and decision trees. Pattern Recognit Lett **29**(7), 918–924 (2008)
41. T Bujlow, T Riaz, J Pedersen, Proceedings of the International Conference on Computing, Networking and Communications (ICNC), in *A method for classification of network traffic based on C5. 0 Machine Learning Algorithm* (IEEE, Maui, 2012), pp. 237–241
42. H Eid, A Hassanien, T Kim, S Banerjee, *Proceedings of the 1st International Conference on Advances in Security of Information and Communication Networks (SecNet). Linear correlation-based feature selection for network intrusion detection model* (Springer, Cairo, 2013), pp. 240–248
43. M Tavallaee, E Bagheri, W Lu, A Ghorbani, Proceedings of the 2nd Symposium on Computational Intelligence for Security and Defence Applications(CISDA), in *A detailed analysis of the KDD CUP 99 data set* (IEEE, Ottawa, 2009), pp. 1–6
44. J Yu, H Kang, D Park, H Bang, D Kang, An in-depth analysis on traffic flooding attacks detection and system using data mining techniques. J Syst Architect **59**(10), 1005–1012 (2013)
45. S Rastegari, P Hingston, C Lam, Evolving statistical rulesets for network intrusion detection. Appl Soft Comput **33**, 348–359 (2015)