COMPUTERS & SECURITY 65 (2017) 135–152

Available online at www.sciencedirect.com

ScienceDirect

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

Computers & Security

A survey of intrusion detection systems based on ensemble and hybrid classifiers

Abdulla Amin Aburomman *, *Mamun Bin Ibne Reaz*

Department of Electrical, Electronic & Systems Engineering, Faculty of Engineering & Built Environment, National University of Malaysia, 43600 UKM Bangi, Selangor, Malaysia

ARTICLE INFO

Article history:
Received 20 July 2016
Received in revised form 5 October 2016
Accepted 8 November 2016
Available online 15 November 2016

Keywords:
Ensemble classifiers
Hybrid classifiers
Intrusion detection
KDD 99
Multiclass classifiers
NSL-KDD

ABSTRACT

Due to the frequency of malicious network activities and network policy violations, intrusion detection systems (IDSs) have emerged as a group of methods that combats the unauthorized use of a network's resources. Recent advances in information technology have produced a wide variety of machine learning methods, which can be integrated into an IDS. This study presents an overview of intrusion classification algorithms, based on popular methods in the field of machine learning. Specifically, various ensemble and hybrid techniques were examined, considering both homogeneous and heterogeneous types of ensemble methods. In addition, special attention was paid to those ensemble methods that are based on voting techniques, as those methods are the simplest to implement and generally produce favorable results. A survey of recent literature shows that hybrid methods, where feature selection or a feature reduction component is combined with a single-stage classifier, have become commonplace. Therefore, the scope of this study has been expanded to encompass hybrid classifiers.

1. Introduction

Constructing a good model from a given data set is one of the major tasks in machine learning (ML). Strong classifiers are desirable, but are difficult to find. Training many classifiers at the same time to solve the same problem, and then combining their output to improve accuracy, is known as an ensemble method. When an ensemble, also known as a multi-classifier system, is based on learners of the same type, it is called a homogeneous ensemble. When it is based on learners of different types, it is called a heterogeneous ensemble. Usually, the ensemble's generalization ability is better than a single classifier's, as it can boost weak classifiers to produce better results than can a single strong classifier. Two results published in the 1990s opened a promising new door for creating strong classifiers using ensemble methods. The empirical study in Hansen and Salamon (1990) found that a combination of multiple classifiers produces more accurate results than the best single one, and the theoretical study in Schapire (1990) showed that weaker classifiers can be boosted to produce stronger classifiers. There are two essential elements involved, in the design of systems that integrate multiple classifiers. First, it is necessary to follow a plan of action to set up an ensemble of classifiers with characteristics that are sufficiently diverse. Second, there is the need for a policy for combining the decisions, or outputs, of particular classifiers in a manner that strengthens accurate decisions and weakens erroneous classifications. Section 2 covers some of the most-used methods, regarding the first element, namely: bagging and its variations, boosting and its generalized version AdaBoost, stacking, and, finally, a mixture of competing experts. In section 3, strategies for achieving the

* *Corresponding author.*
E-mail addresses: reoroman@hotmail.com (A.A. Aburomman), mamun.reaz@gmail.com (M.B.I. Reaz).
http://dx.doi.org/10.1016/j.cose.2016.11.004
0167-4048/

second element are described. Section 4 presents an overview of ensemble techniques for intrusion detection systems. In section 5, an exploration of results obtained from machine learning techniques that use different ensemble approaches is given. Finally, concluding remarks and a critical analysis are expressed in section 6.

2. Methods of creating ensemble classifiers

In recent years, an abundance of ensemble-based classifiers has been produced and improved. Nonetheless, a number of these classifiers are variations on just a few well-established algorithms with capabilities that have been comprehensively validated and broadly published. An overview of the most commonly used ensemble algorithms is presented in this section.

2.1. *Bagging*

Breiman's bootstrap aggregating method, or "bagging" for short, was one of the first ensemble-based algorithms, and it is one of the most natural and straightforward ways of achieving a high efficiency (Breiman, 1996). In bagging, a variety of results is produced, using bootstrapped copies of the training data; that is, numerous subsets of data are randomly drawn with replacement from the complete training data. A distinct classifier of the same category is modeled, using a subset of the training data. Fusing of particular classifiers is achieved by the use of a majority vote on their selections. Thus, for any example input, the ensemble's decision is the class selected by the greatest number of classifiers. Algorithm 1 contains a pseudocode for the bagging method.

An approach that is derived from bagging is called the "random forests" classifier. It received its name because it builds a model from a number of decision trees (Breiman, 2001). A means of creating this kind of classifier is by training different decision trees, and randomly varying parameters related to training. As in bagging, those parameters can be bootstrapped copies of the training data; however, in contrast with bagging, they also can be particular feature subsets, which is the practice in the random subspace method.

Another approach that is derived from bagging is called "pasting of small votes." Unlike bagging, pasting small votes was an approach devised to operate on large data sets (Breiman, 1999). Data sets of a large size are partitioned into subsets of a smaller size, which are called "bites," and those bites are used to train different classifiers. Pasting small votes has led to the creation of two variations: the first one, known as Rvotes, generates the data subsets at random; the other, called Ivotes, builds successive data sets, considering the relevance of the instances. Of the two, Ivotes has been shown to yield better outcomes (Chawla et al., 2002), similar to the idea present in the boosting-based methods, by which each classifier directs the most relevant instances for the ensemble part that is in use.

2.2. *Boosting*

It was shown by Schapire, in 1990, that a weak learner, namely an algorithm that produces classifiers that can slightly outperform random guessing, can be transformed into a strong learner, namely an algorithm that constructs classifiers capable of correctly classifying all of the instances except for an arbitrarily small fraction (Schapire, 1990). Boosting generates an ensemble of classifiers, as does bagging, by carrying out resampling of the data and combining decisions using a majority vote. However, that is the extent of the similarities with bagging. Re-sampling in boosting is carefully devised so as to supply consecutive classifiers with the most informative training data. Essentially, boosting generates three classifiers as follows: A random subset of the available training data is used for constructing the first classifier. The most informative subset given for the first classifier is used for training the second classifier, where the most informative subset consists of training data instances, such that half of them were correctly classified by the first classifier and the other half were misclassified. Finally, training data for the third classifier are made of instances on which the first and second classifiers were in disagreement. A three-way majority vote is then used, to combine the decisions of the three classifiers.

In 1997, Freund and Schapire presented a generalized version of the original boosting algorithm called "adaptive boosting" or "AdaBoost" for short. The method received that name from to its ability to adapt to errors related to weak hypotheses, which are obtained from WeakLearn (Freund). AdaBoost.M1 and AdaBoost.R are two of the most frequently used variations of this category of algorithms, because they are suitable for dealing with multi-class and regression problems, respectively. AdaBoost produces a set of hypotheses, and then uses weighted majority voting of the classes determined by the particular hypotheses in order to combine decisions. A weak classifier is trained to generate the hypotheses, by drawing instances from a successively refreshed distribution of the training data. The updating of the distribution guarantees that it will be more likely to include in the data set for training the subsequent classifier examples that were wrongly classified by the preceding classifier. Thus, the training data of successive classifiers tend to advance toward increasingly hard-to-classify instances.

Algorithm 1 Bagging

Input: I (a classifier inducer), T (# of iterations), S (Data set for training), N (subset size).

Output: C_t; $t = 1, 2, .., T$

1: $t \leftarrow 1$
2: **repeat**
3: $S_t \leftarrow$ Subset of N instances taken, with replacement, from S.
4: Create Classifier C_t by using I on S_t.
5: t ++
6: **until** $t > T$.

Algorithm 2 AdaBoost

Input: I (a weak classifier inducer), T (# of iterations), S (Data set for training), N (subset size).

Output: C_t, α_t; $t = 1, 2, .., T$

$t \leftarrow 1$
$D_1(i) \leftarrow 1/m$; $i = 1, 2, .., m$
repeat
  Create Classifier C_t by using I and the distribution D_t.
  $\epsilon_t \leftarrow \sum_{i:C_t(x_i)\neq y_i} D_t(i)$
  if $\epsilon_t > 0.5$ **then**
    $T \leftarrow t - 1$
    exit Loop
  end if
  $\alpha_t \leftarrow \frac{1}{2} ln \frac{1-\epsilon_t}{\epsilon_t}$
  $D_{t+1}(i) \leftarrow D_t(i) \cdot e^{-\alpha_t y_t C_t(x_i)}$
  Normalize D_{t+1} so that it becomes a distribution
  t ++
until $t > T$.

Pseudocodes for AdaBoost, and the similar AdaBoost.M1, are shown in algorithm 2 and algorithm 3, respectively.

2.3. *Stacking*

Some instances are very likely to be misclassified, because it can happen that they are in the close neighborhood of the decision boundary, and, therefore, usually are placed on the wrong side of the boundary determined by the classifier. On the other hand, there can be instances that are likely to be classified well, as a result of being on the correct side and far away from the corresponding decision boundaries. This prompts the following question: can it be learned whether specific classifiers consistently perform correct classifications, or whether they consistently classify specific examples incorrectly? Said another way, if there is an ensemble of classifiers working with a data set taken from an unknown-but-fixed distribution, can we define a correspondence between the decisions of those classifiers and their correct classes? The idea behind Wolpert's stacking generalization is that the outputs of an ensemble of classifiers serve as the inputs to another, second-level meta-classifier, which has the purpose of learning the mapping that relates the ensemble outputs with the real true classes (Wolpert, 1992).

2.4. *Mixtures of competing experts*

Mixtures of competing experts (Jacobs et al., 1991) is a technique that approaches the problem in a way similar to stacking. In this method, the ensemble is created using a set of classifiers $C_1, \ldots, C_T$, followed by a second-level classifier C_{T+1}, which has the purpose of assigning the weights that a subsequent combiner requires for fusing decisions. An important characteristic is that the combiner is not generally a classifier, but is a plain combination rule, as is, for instance, random selection (from a weight distribution), weighted majority, or winner-takes-all. Although the combiner might not be a classifier, the set of weights that the combiner uses is selected by a second-level classifier, commonly a neural network that is called a gating network. The training method for the gating network is either a standard back-propagation based on gradient descent, or, more frequently, the expectation maximization (EM) algorithm (Jordan and Jacobs, 1994; Jordan and Xu, 1995). Whatever is the case, the actual training data instances constitute the

Algorithm 3 AdaBoost.M1

Input: I (a weak classifier inducer), T (# of iterations), S (Data set for training), N (subset size).

Output: C_t, β_t; $t = 1, 2, .., T$

$t \leftarrow 1$
$D_1(i) \leftarrow 1/m$; $i = 1, 2, .., m$
repeat
  Create Classifier C_t by using I and the distribution D_t.
  $\epsilon_t \leftarrow \sum_{i:C_t(x_i)\neq y_i} D_t(i)$
  if $\epsilon_t > 0.5$ **then**
    $T \leftarrow t - 1$
    exit Loop
  end if
  $\beta_t \leftarrow \frac{\epsilon_t}{1-\epsilon_t}$
  $D_{t+1}(i) \leftarrow D_t(i) \cdot \begin{cases} \beta_t & C_t(i) = y \\ 1 & otherwise \end{cases}$
  Normalize D_{t+1} so that it becomes a distribution
  t ++
until $t > T$.

inputs to the gating network, which is in contrast with the stacking approach, which uses the decisions of first-level or base classifiers. Therefore, the combination rule uses weights that are instance-specific, devising a dynamic combination rule. The mixture of competing experts technique can be, accordingly, categorized as a classifier selection algorithm. Particular classifiers specialize in a region of the feature space, and the purpose of the combination rule is to select the most suitable classifier. Or, alternatively, classifiers can be balanced according to their expertise, with respect to the instance x. The weights may be used by the pooling or combining system in various ways: a single classifier may be selected, if it exhibits the highest weight; or a weighted sum of classifier outputs may be computed for each class, and the class with the highest weighted sum may be chosen. That last strategy is applicable, if the classifier outputs are continuous-valued for each class.

3. Methods that combine classifiers

The practice of combining classifiers is the second fundamental element present in ensemble schemes. This approach uses combination rules that are usually categorized according to the following criteria: (i) combination rules that are trainable vs. those that are non-trainable; or, alternatively, (ii) class labels vs. class-specific applicable combination rules. An independent algorithm establishes the parameters required by the combiner, which are commonly called "weights," in the case of trainable combination rules. An example of this category of methods is the EM algorithm used in the mixture of competing experts model. In the trainable combination rules, the parameters are generally instance-specific, and, for this reason, are known as dynamic combination rules. In contrast, in the case of non-trainable combination rules, the training is not independent; instead, it is incorporated to the training of the ensembles. Weighted majority voting falls into this category of non-trainable rules, as discussed below, given that the weights are directly obtained when the classifiers are created. According to the other taxonomy, class labels having applicable rules that solely require the classification decision (i.e., one of ω_j, $j = 1, \ldots, C$) are opposed to those having inputs consisting of continuous-valued outputs produced by particular classifiers. Generally, what these values represent is to what extent the classifiers support each class, and, consequently, they can be used to estimate class-conditional posterior probabilities $P(\omega_j|x)$. Two conditions are required for that last statement: (i) the values have to be properly normalized, so that they add up to 1 considering all classes; and (ii) the training data used by the classifiers are required to be sufficiently dense.

There exist many models that correspond to this category: MLP and RBF networks are typical examples. Those two models produce continuous-valued outputs that are commonly used as posterior probabilities, although the second required condition concerning sufficiently dense training data often is not met. This paper focuses on the second taxonomy: first, combination rules applicable to class labels are analyzed, and subsequently the methods that fuse class-specific continuous outputs are considered.

3.1. Methods that combine class labels

For the ideas presented in this section, the assumption is made that the classifier outputs consist of only the class labels. The decision that is produced by the t^{th} classifier is designated as $d_{t,j} \in (0, 1), t = 1 \ldots, T$, where $j = 1, \ldots C, T$ is the number of classifiers and C is the number of classes. The combined decision will produce $d_{t,j} = 1$, if the t^{th} classifier decides for class ω_j, and $d_{t,j} = 0$ otherwise.

3.1.1. Variants on majority voting

Majority voting ensemble methods can be categorized into three versions, with different strategies of choosing the class. In the different strategies, the decisions are taken as follows: (i) the class is assigned with the agreement of all the classifiers, and this approach is known as "unanimous voting;" (ii) the decision is made, if the number of classifiers agreeing in one class is at least one more than half of the total number of classifiers, which is commonly known as "simple majority;" and, finally, (iii) the class assigned is that which receives the majority of the votes, without the condition that the sum of votes is greater than any percentage of the models, and this way of deciding is known as "plurality voting" or "majority voting" without any other adjective. The output of the ensemble, in the last category of plurality voting, can be outlined with the following proposition: select class w_j, whenever the following is true:

$$\sum_{t=1}^{T} d_{t,J} = \max_{j=1}^{C} \sum_{t=1}^{T} d_{t,j} \tag{1}$$

3.1.2. Weighted majority rule

The plurality voting mechanism can be surpassed, in terms of overall performance, if a strategy is devised with the knowledge that some of the experts are better than others at making decisions. The decisions of those better qualified experts can be taken into account, using a larger weight than the others. "let us designate the decision that is produced by the t^{th} classifier upon class ω_j as $d_{t,j}$, and establish that, if the t^{th} classifier selects ω_j, then $d_{t,j} = 1$, otherwise $d_{t,j} = 0$." Then, accordingly, class ω_j is chosen by this method of weighted majority rule, if the combination of the decisions made by the classifiers satisfies the following:"

$$\sum_{t=1}^{T} w_t d_{t,j} = \max_{j=1}^{C} \sum_{t=1}^{T} w_t d_{t,j} \tag{2}$$

Other schemes that combine class labels, and that are worth mentioning here, are the behavior knowledge space (BKS) (Huang and Suen, 1993) and Borda count (Van Erp and Schomaker, 2000).

3.2. Methods that combine continuous outputs

There are also classifiers that provide a continuous output for each class. In those schemes, that output represents how much that class is endorsed by the classifier. That value is, in some cases, taken as a predicted value for the respective class posterior or revised probability. The requirements for accepting this type of continuous output value as an estimate of the

posterior probability are that the sum of the values corresponding to all classes, once normalized, must add up to 1; and that the classifier deals with sufficiently dense accessible data for training. The normalization usually selected for this purpose is the softmax function (Mikolov et al.).

4. Overview of ensemble techniques

In the literature, one can see the gradual development and implementation of a wide variety of anomaly detection systems based on various machine learning techniques. Many studies have implemented single-stage learning algorithms, such as artificial neural networks (ANN), genetic algorithms (GA) and support vector machines (SVM). However, systems based on a combination of several methods, such as hybrid or ensemble systems, have been common as well. This section presents an overview of such approaches for intrusion detection systems. The overview is accompanied by an analysis of voting-based ensemble techniques in other fields of research.

Early research by Dietterich (2000) and Miranda Dos Santos (2008) showed, both theoretically and empirically, that ensembles are superior to single-component classifiers, in terms of classification accuracy. With the implementation of multiple base classifiers, the overall error rate of an ensemble can be reduced, provided that each base classifier is better than a random guess, namely that the overall accuracy of the base classifier is over 50%. The advantages of ensemble classifiers are particularly evident in the field of intrusion detection, since there are many different types of intrusions, and different detectors are needed to detect them (Axelsson, 2000). Moreover, if one classifier fails to detect an attack, then another classifier in the ensemble should detect it (Lee et al., 2000). Based on an ensemble's structure, two general approaches may be distinguished: (i) homogeneous ensembles, where all classifiers in the ensemble are generated with the same technique; and (ii) heterogeneous ensembles, which utilize diverse base classifiers. Ensemble techniques like bagging and boosting are often used to generate homogeneous ensembles, whereas stacking and voting can be used to produce heterogeneous ensembles.

Active research of ensemble-based systems by Chen et al. (2014) and Kumar and Kumar (2013) raises several open questions:

- How should suitable base components for an ensemble be created?
- How should it be decided upon which base classifiers one should rely?
- How should the decisions of base classifiers be combined into a final decision?

4.1. *Homogeneous ensembles for IDS*

In general, homogeneous ensembles can be viewed as a simple and effective way of extending the classification hypotheses of a single classification algorithm by creating several variations of that classifier. Although there are numerous ensemble methods by which this can be achieved, the core principles are the same: the aggregation of several relatively simple decision rules should lead to a more sophisticated and reliable final decision. Usually, the selected classifier is trained with different training subsets, at various stages of ensemble development. As a result, the classifier analyzes the problem from different perspectives, and, each time, aggregates the knowledge gained towards the definition of an ensemble classification hypothesis. This section presents an overview of several homogeneous ensemble techniques used in IDS construction. The description of the works will be organized according to the ensemble methods, and special attention will be given to the transition from one scheme to another. Table 1 contains relevant characteristics of the homogeneous methods related to IDS presented in this section, for comparison purposes.

In the boosting group of algorithms, Folino et al. (2010) proposed a method for a distributed intrusion detection that used genetic programming to generate decision-tree classifiers. These classifiers were then combined into an ensemble using AdaBoost.M2, a variant of AdaBoost. The KDD 99 data set was used to evaluate the proposed system. Experimental results showed that the proposed approach was comparable to the top two entries to the KDD Cup 99. Additionally, this technique was shown to be suitable for distributed intrusion detection. Also in this group, Gudadhe et al. (2010) used boosting to combine a family of decision trees into an ensemble. They presented an experimental study, in which the approach they developed was compared to naïve Bayes, kNN, the winning entry from KDD Cup 99 (Pfahringer, 2000), eClass0, and eClass1 (Angelov and Zhou, 2008). They reported that their approach out-performed the other algorithms on the KDD 99 data set. In contrast with the previous work, this implementation was capable of detecting all kinds of attacks.

Among other methods that used boosting together with another technique, Bahri et al. (2011) introduced a hybrid approach, based on an ensemble method called Greedy-Boost. In their experiments, they compared the precision and recall of AdaBoost, C4.5, and Greedy-Boost, for classification of the KDD 99 data set. Reported results indicated that Greedy-Boost outperformed the other algorithms in terms of the precision, even for probe, U2R, and R2L attacks. This method was good at detecting rare attacks, and also lowered average cost, but was not tested on unseen attacks. In another work, Syarif et al. (2012) implemented bagging, boosting, and stacking ensemble methods, to solve the intrusion detection problem. The primary objective of their research was to improve classification accuracy and to reduce false positive rates, for classification of the NSL-KDD data set. The bagging and boosting ensembles were constructed with four traditional classification algorithms: naïve Bayes, J48 (decision trees), JRip (rule induction), and IBK (nearest neighbor). Additionally, heterogeneous ensembles were constructed using a stacking strategy, where each of four algorithms was used in turn to perform meta-level classification. Their approach achieved an accuracy of more than 99%, at detecting known intrusions. However, for new types of intrusions, the accuracy rate was only 60%. The use of homogeneous ensembles created with bagging and boosting showed no significant gain in accuracy. On the other hand, the heterogeneous ensemble set up with stacking led to a significant reduction (46.84%) in false positive rates.

Table 1 – Comparison of methods for homogeneous ensembles.

Ensemble method	Pre-processing	Classifiers and Task	Pros	Cons	Data set
(Folino et al., 2010) Boosting	Not used	GP – to classify as "normal" vs. four types of attacks	Suitable for distributed intrusion detection	Cannot be used as general-purpose	Full KDD 99
(Gudadhe et al., 2010) Boosting	Not used	DT – to classify as "normal" vs. four types of attacks	Used to detect all kinds of attacks	(ii) Was not tested on new (unseen) attacks (ii) Experimental results not available	KDD 99 subset
(Bahri et al., 2011) Greedy-Boost	Not used	C4.5 - to classify as "normal" vs. four types of attacks	(i) Good at detecting rare attacks, and (ii) lower average cost	Was not tested on new (unseen) attacks	KDD 99 subset
(Syarif et al., 2012) Bagging, boosting, and stacking	Not mentioned	NB, J48, JRip, and iBK – to classify as "normal" vs. "anomaly"	Good at detecting known intrusion types	(i) The system could not detect novel attacks. (ii) The use of bagging and boosting homogeneous ensembles was unable to significantly improve the accuracy. (iii) The method was insufficient for implementation in the intrusion detection field.	NSL-KDD subset
(Gaikwad and Thool, 2015) Bagging	GA feature selection	DT – to classify as "normal" vs. "anomaly"	Had a reduced model-building time	Was not tested on new (unseen) attacks	NSL-KDD subset
(Lin et al., 2012) Majority voting	Not used	SVM – to classify as "normal" vs. four types of attacks	Good at detecting R2L and Prob known attacks	Was not tested on new (unseen) attacks	KDD 99 subset
(Kumar and Kumar, 2013) Majority voting	Not used	NB – to classify as "normal" vs. four types of attacks	A generalized classification approach that is applicable to the problems of most any field	Requires a long time to compute fitness functions for various generations	KDD 99 and ISCX 2012 subsets
(Malik et al., 2011) Majority voting	Particle swarm optimization (PSO) feature selection	RF – to classify Prob attacks	Good for Prob detection	Samples used for training and testing were from the same distribution.	KDD 99 subset
(Bukhtoyarov and Zhukov, 2014) GP	Not used	ANN – to classify Prob attacks	Good for detection of PROBE attacks	(i) Was not tested on new (unseen) attacks, and (ii) not as high of accuracy as other approaches	KDD 99 subset
(Masarat et al., 2014) Fuzzy combiner	Roulette wheel algorithm based on gain ratios for selecting features, and random forests for evaluating features	Decision tree J48 - to classify as "normal" vs. four types of attacks	Reduce computation cost	(i) Cannot be used in real-time (ii) Incomplete experimental results	Full KDD 99

Working with a bagging scheme, Gaikwad and Thool (2015) conducted experiments using the NSL-KDD data set, in which six different binary classifiers were compared: partial decision tree classifiers (PART), naïve Bayes, C4.5, a bagged family of PART base classifiers, a bagged family of naïve Bayes classifiers and a bagged family of C4.5 classifiers. In all cases, GA was used to reduce the dimensionality of the input feature space from 41 to 15. Surprisingly, the bagged PART ensemble performed worse than C4.5 without bagging. C4.5 had a classification accuracy of 79.08%, compared to 78.37% for the bagged PART ensemble. Also, the C4.5 model could be trained nine times faster. In the other two cases of bagging, the bagged ensembles were no better than the individual base classifiers, in terms of classification accuracy as well as training time. The method implemented in this work reduced the model-building time, but the approach was not tested on unseen attacks.

A group of methods used the majority voting scheme. Lin et al. (2012) presented an SVM ensemble method based on rotation forest. The results of the classifiers were combined using majority voting. The KDD 99 data set was used to test the performance of the method. Their results showed that an ensemble of two-layer SVM based on rotation forest achieved better accuracy on R2L and probe attacks. However, the method was not tested on unseen attacks. Using the scheme of majority voting as well, Kumar and Kumar (2013) developed an evolutionary approach for IDS based on multi-objective GA, where the archive-based microgenetic algorithm 2 (AMGA2) was used to find optimal trade-offs for multiple criteria, and, in order to integrate the decisions of base classifiers, majority voting was used. The approach applied a generalized classification applicable to any field, but there was a high computational cost in obtaining fitness functions. And, finally, in this group, Malik et al. (2011) presented a classifier based on binary particle swarm optimization (BPSO), and random forests for classification and detection of probe attacks in networks. The performance was validated using the KDD 99 data set. The method performed well for the probe attacks, but with the shortcoming that samples in training and testing were from the same distribution.

A scheme that was built using techniques from genetic programming is also part of this overview. Bukhtoyarov and Zhukov (2014) developed a probabilistic approach to designing base neural network classifiers, called probability based generator of neural network structures (PGNS). The aggregation of neural network classifiers was performed with genetic programming-based ensembling (GPEN). GPEN utilized genetic programming operators, to find an optimal function for combining the base classifiers into an ensemble. The research was conducted on the KDD 99 data set, where the goal was to distinguish between probe and non-probe attacks, based on nine of the 41 attributes. They compared the results with those published in other research (Malik et al., 2011). The results that were obtained using their approach showed better detection accuracy of probe attacks than almost all the competing approaches included in Malik et al. (2011). The only approach that had better detection accuracy and fewer false positives was the PSO-RF approach. This method is particularly good in detecting probe attacks, but it was not tested on unseen attacks. Another disadvantage is that its accuracy is not as high as other techniques.

To conclude discussion of homogeneous ensembles, Masarat et al. (2014) implemented a fuzzy combiner, as a method of obtaining an ensemble decision from multiple decision tree classifiers. The experimental procedure was conducted on the full KDD 99 data set, and a decision tree classifier (J48) was used as a base algorithm. The pre-process involved the roulette wheel algorithm, based on the gain ratios for selecting features, where each decision tree was generated with a unique subset of features. Finally, the decisions of all of the trained classifiers were weighted and combined in a fuzzy ensemble classifier. The authors reported the accuracy to be nearly 93%, based on 15 selected features. This method has the advantage of solving the computing time limitation, but cannot be used in real-time.

It can be observed in Table 1 how homogeneous schemes employed in IDS have progressed in recent years. The earliest approaches generally did not include pre-processing, and were aimed at specific, rather than general, purposes. More recent homogeneous configurations of classifiers have improved, in their detection capabilities, especially for probe attacks. Efficiency has also improved, although real-time problems continue to need development, in order to become usable.

4.2. *Heterogeneous ensembles for IDS*

The defining characteristic of heterogeneous ensembles is that the final decision is based on the classification rules of diverse base classifiers. The chief obstacle to creating such ensembles is that each expert in the ensemble employs a particular method to construct its classification hypothesis. To generate heterogeneous ensembles, the output of each base classifier must be interpretable in the same way. There are various strategies for aggregating the classification results into a final decision, and the voting procedure is one of the simplest and easiest methods to implement. In this section, an overview of heterogeneous ensemble classifiers is presented, with particular attention given to methods based on voting and weighted voting strategies. As in the previous section, relevant aspects of methods are presented in Table 2, with the aim of highlighting comparable elements.

First will be considered a group of heterogeneous methods that used majority voting to combine decisions. An early contribution was presented by Mukkamala et al. (2005), which combined decisions made by classifiers of five different kinds, namely: SVM, MARS, ANN (RP), ANN (SCG) and ANN (OSS). The data set used in this work was a subset of DARPA 1998 and the ensemble performance showed better accuracy than individual classifiers. Time cost, mainly due to ANNs, was a shortcoming.

More recently, Govindarajan and Chandrasekaran (2012) proposed a hybrid ensemble method that combined the decisions of diverse classifiers. They implemented a generalized version of bagging and boosting algorithms. Adaptive re-sampling and combining, also called "Arcing," was used to generate different training sets, for two classifiers: radial basis function (RBF) neural network and SVM. In addition, the authors implemented a best-first search (BFS), for feature selection. The final decision was reached by majority voting. An experimental procedure, conducted on the NSL-KDD data set, demonstrated that a hybrid approach was more effective than a single classifier. The reported classification accuracy of the RBF-SVM ensemble was 85.17%.

Table 2 – Comparison of methods for heterogeneous ensembles.

Ensemble method	Pre-processing	Classifiers and Task	Pros	Cons	Data set
(Mukkamala et al., 2005) Majority voting	Not used	ANN, SVM and MARS – to classify as "normal" vs. four types of attacks	Ensemble outperforms single classifiers	ANNs take a long time to train	DARPA 1998 subset
(Govindarajan and Chandrasekaran, 2012) Majority voting	BFS feature selection	RBF and SVM – to classify as "normal" vs. four types of attack	Easy to implement	(i) Not applicable for real-time detection, (ii) low detection accuracy, and (iii) was not tested on new (unseen) attacks	NSL-KDD subset taken from one set
(Meng and Kwok, 2013) Majority voting	Manual selection of eight features	SVM, DT, and kNN – to classify as false vs. true alarms using Snort software	Could reduce false alarm rate	Not applicable for real-time	DARPA 99 subset and lab-generated data set
(Haq et al., 2015) Majority voting	Feature selection using three wrapper approaches: best first, genetic search, and ranker search	BN, NB, and J48 decision trees – to classify as "normal" vs. "anomaly"	(i) Little time was required to build a model, and (ii) the ensemble achieved better results than single methods.	Was not tested on new (unseen) attacks	NSL-KDD subset taken only from training set
(Gu et al., 2008) Weighted averaging	PCA and ICA feature extraction	SVM – to classify as "normal" vs. four types of attack	Good at decreasing false negative errors	Ensemble could not achieve high accuracy	KDD 99 subset taken from one set
(Syarif et al., 2012) Stacking	Not mentioned	IBK, J48, JRip, and NB – to classify as "normal" vs. "anomaly"	Ensemble set up with stacking led to a significant reduction in false positives	Failed to detect novel intrusions	NSL-KDD subset taken from one set
(Chan et al., 2005) 1. Majority voting, 2. weighted majority voting, 3. NB stacking, 4. Dempster-Schafer combination, 5. averaging, and 6. ANN stacking	Not used	MLP, RBF-ANN, and SVM – to classify as "normal" traffic vs. six kinds of DoS attacks	(i) ANN stacking achieved the best result. (ii) Good at detecting DoS attacks	Requires more samples and detection of different types of attacks	KDD 99 subset taken only from training set
(Borji, 2007) 1. Majority voting, 2. averaging, and 3. belief measurement	Not used	ANN, SVM, C4.5, and kNN – to classify as "normal" vs. four types of attacks	Good for known attacks	Not tested on novel attacks	DARPA 1998 subset
(Tama and Rhee, 2015) (i) Average of probability, and (ii) majority voting	PSO and correlation-based feature selection	C4.5, random forest, and CART – to classify as "normal" vs. "anomaly"	The proposed ensemble method performed better than any other single classifier.	(i) The performance of ensemble classifiers degraded, when the number of PSO particles increased. (ii) Was not tested on new (unseen) attacks	NSL-KDD subset taken only from training set

Likewise, Meng and Kwok (2013) experimented with both single and ensemble classifiers composed of J45, *k*NN, and SVM, for classification of the 1998 DARPA intrusion detection evaluation data set. They found that an ensemble of all three classifiers, based on majority voting, marginally out-performed all other combinations.

To close this group, Haq et al. (2015) developed an IDS in three phases: (*i*) a hybrid approach to feature selection, (*ii*) classification with base classifiers, and (*iii*) deployment of a majority voting strategy to form the final decision. The feature selection process was based on three methods: BFS, genetic search (GS), and ranking search (RS). The final set of features was derived, by combining the results from all three feature selection algorithms, where the features most commonly chosen by all three algorithms were propagated to the last set. The classification at the second stage was performed by three classification algorithms: naïve Bayes (NB), Bayesian network (BN), and J48 (classification trees). The experimental procedure was performed on the NSL-KDD data set. Although the proposed approach showed improved computational efficiency, it classified data with worse accuracy, when compared to a majority voting ensemble based on only RS feature selection.

There is also a category of works that used an approach that is different than majority voting. Gu et al. (2008) developed a weighted averaging ensemble for binary classification (normal vs. attack) of the KDD 99 data set. Two base classifiers were created with the SVM algorithm and two data-reduction methods: principal component analysis (PCA) and independent component analysis (ICA). The ensemble weights were generated with a multi-objective genetic algorithm (NP-GA), where the goal was to obtain the pareto-optimal solution for minimization of false positive and false negative rates. This experimental study reported an improvement in classification accuracy, for the weighted average of two base classifiers. In the previous section, work by Syarif et al. (2012) was presented, because of the inclusion of bagging and boosting in their study. They also evaluated stacking, as a method for combining classifiers' decisions, with a NSL-KDD subset, finding that it was able to reduce the false positive rate, but with a long execution time.

A third group of works experimented with different approaches for combining decisions. Chan et al. (2005) compared several approaches to generating an ensemble-based IDS:

- Majority voting
- Weighted majority voting
- Stacking with NB
- Dempster–Schafer combination, as defined in Rogova (1994)
- Averaging posterior probability
- Stacking with ANN

The examined ensemble methods were based on three classification approaches: multi-layer perceptron (MLP), radial basis function neural network (RBF-ANN), and support vector machines. The experimental procedures were conducted on the KDD 99 data set, where the goal was to identify the occurrence of denial of service (DoS) attacks. The authors reported that the best results were achieved with an ANN stacking ensemble, followed closely by a Dempster–Schafer combiner.

Also in this category of works that experimented with combining several schemes, Borji (2007) presented an analysis of three methods for generating a heterogeneous ensemble: majority voting, averaging of posterior probabilities, and belief measurement based on cross-validation results. An experimental procedure was developed for multi-class classification of examples from the 1998 DARPA data set, and the ensemble decision was based on output from four base classifiers: ANN, SVM, C4.5 (decision trees), and *k*NN. All of the ensemble methods performed much better than any of the base classifiers, and the best performance was achieved by a voting ensemble based on belief measurement.

In another work, Tama and Rhee (2015) performed a binary classification (normal vs. attack) of the NSL-KDD data set, with ensembles based on majority voting and the averaging of posterior probabilities. Additionally, they developed a hybrid feature selection method, based on particle swarm optimization and correlation-based feature selection (PSO-CFS), to pre-process the training and testing data. The ensembles were comprised of three decision tree algorithms: C4.5, random forest, and CART. Their experimental results indicated that the best performance was achieved by an ensemble based on the averaging of posterior probabilities. However, it is worth noting that they were able to obtain similar results with boosting of the C4.5 classifier.

In this set of heterogeneous ensemble methods applied to IDS, it is noteworthy that the schemes that employed a straightforward strategy for combining decisions, such as majority voting, have been incorporating complex arrangements for pre-processing, in order to improve accuracy and other measurements of performance. The variety of classification approaches explored also has increased. Other works have dealt with the challenge by producing a diversity of courses of action, in order to find the best selection for particular classifiers as well as for the ensembles. Detection of attacks, reduction of false alarms, and reduction of response times are among the progressive improvements. Issues with novel intrusions or untested examples are among the aspects that still require research and development.

4.3. *Heterogeneous ensembles based on voting applied to other domains*

In order to ensure that all available options have been included, the scope of this literature review has been broadened to include heterogeneous ensemble techniques from other research fields. This analysis of current activities in ensemble research should reveal possibilities for improvements in the construction of ensemble classifiers based on multiple learning algorithms.

Jankowski and Grabczewski (2005) made an extensive comparison of several ensemble methods, namely:

- Majority voting (MV)
- Majority voting, based on global competence (GC-MV)
- Weighted majority voting, based on posterior probability (WMV)
- Weighted majority voting, based on local competence (LC-WMV)

- Weighted majority voting, based on weighted local competence (WLC-WMV)
- Weighted majority voting, based on global competence (GC-WMV)
- Weighted majority voting, based on cross-validation local competence (CV-LC-WMV)
- Weighted majority voting, based on cross-validation weighted local competence (CV-WLC-WMV)
- Weighted majority voting, based on cross-validation global and local competence (CV-GLC-WMV)
- Weighted majority voting, based on cross-validation global and weighted local competence (CV-GWLC-WMV)
- Winner takes all, based on local competence (LC-WTA)
- Winner takes all, based on cross-validation local competence (CV-LC-WTA)
- Winner takes all, based on cross-validation global and local competence (CV-GLC-WTA)

All generated ensembles from their decisions were based on the output of five classifiers: *k*NN, SSV tree, NB, SVM, and linear SVM. The experimental analysis was based on 17 data sets from the UCI repository. The authors reported that more accurate and stable classifiers result from augmenting weighted majority voting ensembles with weights based on local and global competence.

Kuncheva and Rodríguez (2014) compared four heterogeneous ensemble techniques:

- Majority voting
- Weighted majority voting
- Recall combiner (REC)
- Naïve Bayes combiner (NBC)

The novel REC approach was developed based on a WMV strategy. The common weighting scheme, where a single weight is used to measure the reliability of the classifier, was replaced with a more complex weighting scheme, where the reliability of each classifier was measured for each class in the data set. Similarly, the NBC ensemble technique was an extension of the REC approach, where the prior probability for each class in the data set also was taken into account. The authors remarked that each increase in the complexity of the weighting scheme required the introduction of an additional learning stage at the ensemble level. The experimental procedure, conducted on 73 benchmark data sets, implied that there was no definitive best approach, among the four analyzed approaches.

Tahir et al. (2012) compared several methods of multi-class classification with heterogeneous ensembles, based on both weighted and unweighted ensemble models. The ensembles were created with five base classifiers: RaKEL (Tsoumakas et al., 2009), ECL (Read et al., 2011), CLR (Fürnkranz et al., 2008), MLKNN (Zhang and Zhou, 2007), and IBLR (Cheng and Hüllermeier, 2009). The authors also examined five ways of generating an ensemble:

- Averaging the posterior probabilities
- Weighted averaging of posterior probabilities
- Weighted majority voting, based on five-fold cross validation
- Weighted majority voting, based on Dudani's rule (Valdovinos and Sánchez, 2009)
- Weighted majority voting, based on Shepard's rule (Valdovinos and Sánchez, 2009)

That comparative study examined the performance of implemented methods on six popular multi-class data sets from various areas of research. Although the experimental results varied for each data set, the authors reported that ensembles based on averaging the posterior probabilities most often produced favorable results.

Toman et al. (2012) introduced a novel approach to generating weight coefficients for heterogeneous ensembles, with classification output representing spatial coordinates. Furthermore, the authors compared their approach, called generalized weighted majority voting (GWMV), with three other popular ensemble techniques:

- Majority voting
- Weighted majority voting, based on posterior probabilities
- Weighted majority voting, based on logarithmic scaling of posterior probabilities (log WMV)

The developed approach was implemented to spatially locate the optic disc (OD) in retinal images. The weighting scheme in GWMV was extended, by the inclusion of a geometric component that measured the relative distance in an output of various OD location algorithms. Although this was a problem-specific solution to weight generation, it nevertheless demonstrated the effectiveness of properly selected weight coefficients.

Gu and Jin (2012) defined a heterogeneous ensemble, constructed with linear discriminant analysis (LDA), support vector machines with a linear kernel function (L-SVM), and support vector machines with a radial basis function kernel (RBF-SVM) as the base classifiers. The developed model was used for binary classification of electroencephalograph (EEG) recordings. The authors also proposed a weight construction scheme, based on the assumption that there was a positive correlation between the classification rate of the training data set (based on cross-validation) and the classification rate of the test data set. A single weight coefficient was awarded to each base classifier, according to its performance scores. The ensemble decision was reached with weighted majority voting.

Tsoumakas et al. (2004) examined three different approaches for combining the decision rules of heterogeneous classifiers. A set of 10 base classifiers was deployed: decision tables (DTab), JRip, PART, J48, IBK, K*, NB, SMO, RBF, and MLP. The performance of each base classifier was evaluated with cross-validation, and weights were derived based on classification accuracy. Base classifiers were evaluated with paired *t*-tests, where each classifier was rated by comparing its performance against other classifiers in the ensemble. The significance score was computed based on paired *t*-test results. Three strategies for base classifier selection were suggested: (*i*) one or more classifiers with the highest significance score were used in making the final decision, and, if there was more than one classifier, then weighted majority voting was used to combine them; (*ii*) several classifiers with similar significance scores were selected and combined with weighted majority voting; and (*iii*) three classifiers with the highest significance score were used to create a majority voting ensemble.

The authors, however, did not obtain the expected results, for experiments conducted on 40 data sets selected from the UCI repository. They established that under-performance of the proposed selection methods, due to an inability to adequately secure the efficiency of base classifiers with cross-validation, was the cause.

Richiardi and Drygajlo (2007) developed three voting strategies, for combining the decisions of multiple classifiers: rigged majority voting (RMV), weighted rigged majority voting (WRMV), and selective rigged majority voting (SRMV). Implemented ensemble methods relied on the posterior probability made by each classifier, which estimated the likelihood of a given observation's being a member of some class. Consequently, the simple voting procedure (RMV) was modified by a measure of certainty of a classifiers decision, made by that classifier. The RMV used the classifier's posterior probability in place of the weight coefficient; however, that approach was also extended to a WRMV with the introduction of actual weights, based on 10-fold cross-validation results for each classifier. A third ensemble strategy (SRMV) was based on a selection method, where only the classifier with the highest cross-validation reliability made the final decision. The proposed approach was developed for applications in the field of biometric authentication. The experimental results, conducted on two signature modality data sets, were based on three classification algorithms: local features Gaussian mixture model (LGMM), global features global Gaussian model (GGMM), and MLP.

Cheng and Chen (2014) applied the heterogeneous ensemble technique to the face recognition problem. A weighted regional voting-based ensemble of multiple classifiers (WREC) approach was proposed to assign weights to each classifier in the ensemble, based on a facial region's significance. Unlike most approaches to weighted voting, the authors implemented a novel way of computing weights. The leave one out (LOO) strategy was used to determine the significance of each facial region, and to generate the appropriate weight. The final decision was based on five implemented classifiers: PCA, Fisherface, spectral regression dimensional analysis (SRDA), a spatially smooth version of linear discriminant analysis (SLDA), and a spatially smooth version of locality preserving projection (SLPP).

Ye et al. (2013) proposed the WMV-based heterogeneous ensemble for board-level functional fault diagnosis. The multi-class classification problem was solved with two base classifiers – ANN and SVM – where an SVM multi-class framework was developed with a one against rest (OAR) strategy. The final decision was made by aggregating the weighted output from two classifiers. Although this approach was similar to many other studies in the field of ensemble classification, the novelty of the developed ensemble lies in its method of computing weight coefficients, namely the authors used logarithmic scaling of weighted training error to determine the confidence of each deployed classifier. The reported experimental analysis demonstrated empirically that a WMV ensemble can perform better than its base classifiers.

Kausar et al. (2010) presented a weighted majority voting ensemble of binary classifiers, based on PSO-generated weights. The developed ensemble was created with four base classifiers: linear discriminant classifier (LDC), quadratic discriminant classifier (QDC), kNN, and back-propagation neural network (BP), the outputs of which were defined in a binary domain (0 or 1). PSO was used to generate weights, and the final decision was reached with weighted majority voting. A meta-heuristic approach was, therefore, used to find a near-optimal set of weights for which the classification error of the ensemble was minimized. The performance of the defined method was examined, with respect to four UCI repository data sets: Heart, Diabetes, Iris, and Transfusion.

De Stefano et al. (2002) introduced a heterogeneous ensemble based on a WMV strategy, with GA-optimized weights. The authors implemented three base classifiers: BP, learning vector quantization neural network (LVQ), and kNN. An experimental procedure was developed for recognition of handwritten digits, and two feature extraction algorithms were used: central geometrical moments (CGM) and a mean number of pixels belonging to disjointed 8×8 windows that can be extracted from a binary image (MBI). The weights were generated for six resulting methods with GA optimization, where the goal was to minimize the classification error of the WMV ensemble.

Remya and Ramya (2014) implemented a weighted majority voting procedure, with the aim to combine posterior probabilities of three base classifiers into a heterogeneous ensemble. The base classifiers used in their experiment were NB, logistic regression classifier (LRC), and SVM. Unlike most implementations of a WMV strategy, the weighting scheme in this paper was similar to the REC ensemble proposed by Kuncheva and Rodríguez (2014). Class recall was computed as a fraction of correctly classified instances in the validation set. The developed ensemble was tested on a biomedical data classification data set. A similar approach had been previously applied to a radar automatic-target-recognition problem, by Zhang et al. (2011). Output decisions of three base classifiers – maximum correlation classifier (MCC), relevant vector machine (RVM), and SVM – were combined with weight coefficients based on class recall. However, instead of computing weights, Zhang et al. used the posterior probabilities of each classifier for the implemented weighting scheme.

An overview of the most prominent studies, where heterogeneous ensembles based on voting were implemented in research areas that are not related to IDS construction, is presented in Table 3.

5. Other techniques for IDS construction

Given that, for the purposes of this review, the network intrusion simulation has been based on the NSL-KDD data set, it is prudent to explore the results of recent activities in IDS construction for that data set. In addition to ensemble approaches, many machine learning techniques have been applied to IDS development. Some of the most popular approaches belong to the group of hybrid methods, where a classification task is usually decomposed into two stages: (i) feature selection or reduction, and (ii) classification of pre-processed data. The chief advantage of this approach is the significant decrease in computational cost, and many lightweight IDSs have been built along these lines. Also, favorable classification results have ensured that hybrid utilization approaches in IDS construction remain an active research area.

Table 3 – Heterogeneous ensembles based on voting.

Reference	Classifiers	Ensemble Method
De Stefano et al. (2002)	BP, LVQ, and kNN	GA-WMV
Tsoumakas et al. (2004)	DTab, JRip, PART, J48, IBK, K*, NB, SMO, RBF, and MLP	Best
Tsoumakas et al. (2004)	DTab, JRip, PART, J48, IBK, K*, NB, SMO, RBF, and MLP	WMV
Tsoumakas et al. (2004)	DTab, JRip, PART, J48, IBK, K*, NB, SMO, RBF, and MLP	MV
Jankowski and Grabczewski (2005)	kNN, SSV tree, NB, SVM, and L-SVM	MV
Jankowski and Grabczewski (2005)	kNN, SSV tree, NB, SVM, and L-SVM	WMV
Jankowski and Grabczewski (2005)	kNN, SSV tree, NB, SVM, and L-SVM	LC-WMV
Jankowski and Grabczewski (2005)	kNN, SSV tree, NB, SVM, and L-SVM	WLC-WMV
Jankowski and Grabczewski (2005)	kNN, SSV tree, NB, SVM, and L-SVM	GC-WMV
Jankowski and Grabczewski (2005)	kNN, SSV tree, NB, SVM, and L-SVM	CV-LC-WMV
Jankowski and Grabczewski (2005)	kNN, SSV tree, NB, SVM, and L-SVM	CV-WLC-WMV
Jankowski and Grabczewski (2005)	kNN, SSV tree, NB, SVM, and L-SVM	CV-GLC-WMV
Jankowski and Grabczewski (2005)	kNN, SSV tree, NB, SVM, and L-SVM	CV-GWLC-WMV
Jankowski and Grabczewski (2005)	kNN, SSV tree, NB, SVM, and L-SVM	LC-WTA
Jankowski and Grabczewski (2005)	kNN, SSV tree, NB, SVM, and L-SVM	CV-LC-WTA
Jankowski and Grabczewski (2005)	kNN, SSV tree, NB, SVM, and L-SVM	CV-GLC-WTA
Chen and Zhao (2008)	ANN, SVM, C4.5, and kNN	MV
Eleyan et al. (2009)	PCA	WMV
Richiardi and Drygajlo (2007)	LGMM, GGMM, and MLP	RMV
Richiardi and Drygajlo (2007)	LGMM, GGMM, and MLP	WRMV
Richiardi and Drygajlo (2007)	LGMM, GGMM, and MLP	SRMV
Kausar et al. (2010)	LDA, QDA, kNN, and BP	PSO WMV
Tahir et al. (2012)	RaKEL, ECL, CLR, MLKNN, and IBLR	Averaging
Tahir et al. (2012)	RaKEL, ECL, CLR, MLKNN, and IBLR	Weighted averaging
Tahir et al. (2012)	RaKEL, ECL, CLR, MLKNN, and IBLR	Cross-validation WMV
Tahir et al. (2012)	RaKEL, ECL, CLR, MLKNN, and IBLR	Dudani WMV
Tahir et al. (2012)	RaKEL, ECL, CLR, MLKNN, and IBLR	Shepard WMV
Zhang et al. (2011)	MCC, RVM, and SVM	WMV
Gu and Jin (2012)	LDA, L-SVM, and RBF-SVM	WMV
Kuncheva and Rodríguez (2014)	Decision trees (100 base classifiers)	MV
Kuncheva and Rodríguez (2014)	Decision trees (100 base classifiers)	WMV
Kuncheva and Rodríguez (2014)	Decision trees (100 base classifiers)	REC
Kuncheva and Rodríguez (2014)	Decision trees (100 base classifiers)	NBC
Toman et al. (2012)	OD spatial algorithms	MV
Toman et al. (2012)	OD spatial algorithms	WMV
Toman et al. (2012)	OD spatial algorithms	log WMV
Toman et al. (2012)	OD spatial algorithms	GWMV
Ye et al. (2013)	ANN and SVM	WMV
Cheng and Chen (2014)	PCA, Fisherface, SRDA, SLDA, and SLPP	WREC
Remya and Ramya (2014)	NB, LRC, and SVM	WMV

Hota and Shrivas (2014) made a comparative study of various hybrid approaches for both binary (normal vs. attack) and multi-class classifications of the NSL-KDD data set. Each implemented hybrid was based on information gain (IG) feature selection and one of these five classification algorithms: MLP, DTab, C4.5, RF, and REP tree. The authors reported that the best performance was achieved with an IG-RF hybrid classifier.

Pervez and Farid (2014) defined another hybrid approach, based on feature selection and subsequent classification, using the NSL-KDD data set. Feature selection was implemented following the LOO method, and, as a classifier, the authors deployed support vector machines in a one against rest multi-class configuration (OAR-SVM). Their experiment showed that the greatest classification accuracy was achieved with evaluation of 14 selected features.

Enache and Patriciu (2014) have developed a two-stage hybrid approach: (i) feature selection with an IG algorithm and (ii) classification with an SVM method for binary (normal vs. attack) IDS classification. In addition, the authors chose to introduce a meta-optimization based on swarm intelligence algorithms, in order to find the optimal set of classification parameters for SVM. Two approaches were used for optimization of SVM classification parameters: PSO and artificial bee colony (ABC). The reported experimental results, for the NSL-KDD data set, indicated that an ABC-SVM approach achieved slightly higher precision than its counterpart, PSO-SVM.

Eid et al. (2011) proposed a simple hybrid classifier, as a solution to the IDS classification problem. A GA was implemented as a wrapper method for feature selection, in conjunction with a NB classifier. The optimal subset of features was found through minimization of classification error of an NB classifier trained with a given subset of features. In addition to feature selection, the authors implemented the entropy minimization discretization (EMD) method in order to discretize the input data. The experimental results were performed on the NSL-KDD data set, where the whole set was used for training, and the effectiveness of the proposed method was evaluated with 10-fold cross validation.

De la Hoz et al. (2014) implemented a two-component hybrid approach, with a feature selection and classification stage, for

IDS construction. Unlike similar methods, De la Hoz et al. introduced a multi-objective approach to feature selection. Non-dominated sorting genetic algorithm (NSGA) feature selection was implemented, to find a subset of features for which the Jaccard coefficients for each class in the data set were maximized. Classification of the NSL-KDD data set was performed with growing hierarchical self-organizing maps (GHSOM). As in Eid et al. (2011), the whole NSL-KDD data set was used in the training phase, and results were based on 10-fold cross-validation. With a reported accuracy of 99.6%, the approach proposed by De la Hoz et al. performed better than the hybrid classifier defined by Eid et al.

Rastegari et al. (2015) developed an IDS based on genetic algorithm optimization. Binary classification (normal vs. attack) of the NSL-KDD data set, was based on a set of *if–then* rules applied to the selected features. The selection of features, for rule construction and definition condition boundaries, was performed with genetic algorithm optimization, where the goal was to minimize the number of misclassified instances. Additionally, the authors implemented several feature selection methods: correlation-based feature selection (CFS), consistency subset evaluator (CSE), and selection of only real-valued features. Their results indicated that the developed approach was comparable to other single-stage learning methods.

Singh et al. (2015) implemented a binary (normal vs. attack), NSL-KDD classification framework based on an online sequential extreme learning machine (OSELM) classifier. The OSELM classifier was developed to overcome the computational restriction of feed-forward neural networks. The authors defined three classification approaches:

- OSELM classification based on alpha profiling of all features in the data set (Alpha OSELM)
- OSELM classification based on alpha profiling of only the selected features (Alpha FST OSELM)
- OSELM classification based on alpha profiling and beta profiling of only the selected features (Alpha FST Beta OSLEM)

The alpha profiling was applied to the whole NSL-KDD set, to combine two of its features, protocol and service, into an alpha feature. To reduce the training time, beta profiling was also deployed, to remove redundant training pairs from the training set. Feature selection was based on three approaches: filtered subset evaluation (FSE), CFS, and CSE. The authors reported that Alpha FST Beta OSLEM was capable of reducing both dimensionality and training set size, without compromising classification accuracy.

Kanakarajan and Muniasamy (2016) presented an approach based on a greedy randomized adaptive search procedure with annealed randomness (GAR-forest) classifier for both binary (normal vs. attack) and multi-label classification of NSL-KDD. The GAR-forest approach was based on the meta-heuristic greedy randomized adaptive search procedure (GRASP), which was deployed to generate a set of randomized adaptive decision trees. Feature selection was implemented by way of three algorithms: IG, symmetrical uncertainty (SU), and CFS. The authors reported that the GAR-forest classifier was able to out-perform random forest, C4.5, NB, and multi-layer perceptron classifiers. The feature selection also resulted in improvement of classification accuracy.

Table 4 – Popular NSL-KDD classification approaches.

Reference	Feature selection/ Pre-processing	Classification method
Eid et al. (2011)	GA and EMD	NB
Hassanien et al. (2014)	PCA	GA-DT
Enache and Patriciu (2014)	IG	PSO-SVM
Enache and Patriciu (2014)	IG	ABC-SVM
Hota and Shrivas (2014)	IG	MLP
Hota and Shrivas (2014)	IG	DTab
Hota and Shrivas (2014)	IG	C4.5
Hota and Shrivas (2014)	IG	RF
Hota and Shrivas (2014)	IG	REP tree
De la Hoz et al. (2014)	NSGA	GHSOM
Pervez and Farid (2014)	LOO	OAR-SVM
Pajouh et al. (2015)	LDA	NB-*k*NNCF
Rastegari et al. (2015)	CFS	GA classifier
Rastegari et al. (2015)	CSE	GA classifier
Rastegari et al. (2015)	Real-valued features	GA-based classifier
Singh et al. (2015)	Alpha	OSELM
Singh et al. (2015)	Alpha FST	OSELM
Singh et al. (2015)	Alpha FST Beta	OSELM
Kanakarajan and Muniasamy (2016)	IG	GAR-forest
Kanakarajan and Muniasamy (2016)	SU	GAR-forest
Kanakarajan and Muniasamy (2016)	CFS	GAR-forest

Hassanien et al. (2014) presented a multi-layer IDS based on three stages: (*i*) feature extraction with PCA, (*ii*) binary (normal vs. anomalous) classification with a genetic algorithm, and (*iii*) multi-class categorization of anomalous instances with decision trees. The genetic algorithm classification was performed as a set of *if–then* rules, which labeled each observation as either normal network traffic or a network intrusion. The experimental procedure was conducted on the NSL-KDD data set. An analysis of the developed approach found that two-layer classification offered more reliable classification results, when compared to single-stage classifiers. A similar approach was developed by Pajouh et al. (2015). As a feature reduction method, Pajouh et al. implemented an LDA algorithm. The first-tier, binary (normal vs. anomalous) classification was performed with an NB classifier, and anomalous data was classified more precisely in the second tier, with a *k*NNCF (*k*NN with a certainty factor) classifier. The analysis of both methods Hassanien et al. (2014) and Pajouh et al. (2015), indicated that Pajouh et al. had managed to obtain considerably better classification results.

Table 4 provides an overview of popular IDS classification approaches, for research studies based on the NSL-KDD data set.

5.1. *Performance comparison of different methods*

In this section, the results of studies that classified the NSL-KDD data set are compared. In order to compare all of the approaches on an equal footing, the examination has been limited to overall classification accuracy based on the same data set type and size. Only studies that applied the full NSL-KDD data set were used for comparison, as follows:

GAR A decision tree-based classifier (GAR-forest), as defined in Kanakarajan and Muniasamy (2016)
IG-GAR IG for feature selection and a decision tree-based classifier (GAR-forest), as defined in Kanakarajan and Muniasamy (2016)
CFS-GAR CFS and a decision tree-based classifier (GAR-forest), as defined in Kanakarajan and Muniasamy (2016)
SU-GAR SU for feature selection and a decision tree-based classifier (GAR-forest), as defined in Kanakarajan and Muniasamy (2016)
PCA-BFtree PCA for feature selection and a decision tree-based classifier (BFtree), as defined in Hassanien et al. (2014)
PCA-J48 PCA for feature selection and a decision tree-based classifier (J48), as defined in Hassanien et al. (2014)
PCA-NBtree PCA for feature selection and a decision tree-based classifier (NBtree), as defined in Hassanien et al. (2014)
PCA-RF PCA for feature selection and a random forest classifier, as defined in Hassanien et al. (2014)
LDA-NB-kNNCF LDA for feature selection, and a two-tier classifier using NB and *k* nearest neighbor with certainty factor (*k*NNCF), as defined in Pajouh et al. (2015)
RBF-SVM Best first search for feature selection and a majority voting ensemble of RBF neural network and SVMs, as defined in Govindarajan and Chandrasekaran (2012)
LOO-OAR-SVM LOO for feature selection and support vectors machines set in a one against rest multi-class classification framework, as defined Pervez and Farid (2014)

In addition, the classification results obtained by Tavallaee et al. (2009), where the formal introduction of the NSL-KDD data set was made, were also considered.

Table 5 presents the overall accuracies of the above-listed approaches.

Table 5 – Comparison of overall accuracies.

Defined in	Approach name	ACC
Kanakarajan and Muniasamy (2016)	GAR	77.26%
Kanakarajan and Muniasamy (2016)	IG-GAR	78.9%
Kanakarajan and Muniasamy (2016)	CFS-GAR	77.94%
Kanakarajan and Muniasamy (2016)	SU-GAR	77.6%
Hassanien et al. (2014)	PCA-BFtree	68.28%
Hassanien et al. (2014)	PCA-J48	72.88%
Hassanien et al. (2014)	PCA-NBtree	67.01%
Hassanien et al. (2014)	PCA-RF	66.71%
Pajouh et al. (2015)	LDA-NB-kNNCF	82%
Govindarajan and Chandrasekaran (2012)	RBF-SVM	85.17%
Pervez and Farid (2014)	LOO-OAR-SVM	82.68%
Tavallaee et al. (2009)	J-48	81.05%
Tavallaee et al. (2009)	NB	76.56%
Tavallaee et al. (2009)	NBtree	82.02%
Tavallaee et al. (2009)	RF	80.67%
Tavallaee et al. (2009)	RT	81.59%
Tavallaee et al. (2009)	MLP	77.41%
Tavallaee et al. (2009)	SVM	69.52%

The comparison of NSL-KDD classification results, presented in Table 5, suggests that an ensemble classifier based on a majority voting strategy is an effective approach to the construction of intrusion detection systems. Furthermore, the above review of the literature found that employing a weighted majority voting strategy to constitute a final decision from metaheuristically optimized weight coefficients can be a good way to reliably obtain higher results.

6. Conclusion and critical analysis

Although there are many approaches to knowledge extraction, multiple-expert systems remain one of the most active research areas. In particular, pattern classification problems are often solved through the implementation of ensemble-based techniques. An overview of related studies suggested that many such approaches have been successfully employed, in various fields of research. In general, there are many approaches to deploying multiple classifiers. For example, there are methods that mainly reduce variance, such as bagging (Shi et al., 2011) or boosting (Syarif et al., 2012), and methods that reduce bias, such as stacked generalization (Wolpert, 1992). Moreover, there are also methods, such as cascading (Gama and Brazdil, 2000), that generate new attributes based on class probability estimation or delegating (Ferri et al., 2004), where each classifier handles only the part of the training set and the rest is delegated to other classifiers in the ensemble. Despite the rich variety of ensemble techniques, voting-based systems are among the more common ways of combining classifiers. Errors introduced by one classifier can be corrected using right decisions made by the other classifiers, provided that similar performance from all classifiers can be expected. However, if the reliability of each classifier in an ensemble could be estimated beforehand, further improving the overall accuracy of the voting ensemble with the introduction of weight coefficients would be possible. Multiple-classifier systems where the final decision is a combination of weighted base classifiers' decisions are commonly called weighted majority voting ensembles.

This overview of related literature has highlighted two main categories of multiple-classifier systems:

- Homogeneous ensembles, or systems based on a single classification approach
- Heterogeneous ensembles, or systems based on two or more different classification approaches

The deployment of ensemble-based classifiers in the construction of IDSs is illustrated in Fig. 1. The above analysis of other studies in the IDS field has revealed an approximately equal distribution of both homogeneous and heterogeneous ensembles.

Utilization of homogeneous ensembles in IDS construction has been a fruitful ground for research in the past several years. However, parallel analysis of related studies for both approaches, presented in sections 4.1 and 4.2, reveal that the implementation of heterogeneous ensembles in IDSs is somewhat less complete. In Fig. 2, the frequency of development of various heterogeneous ensembles is presented

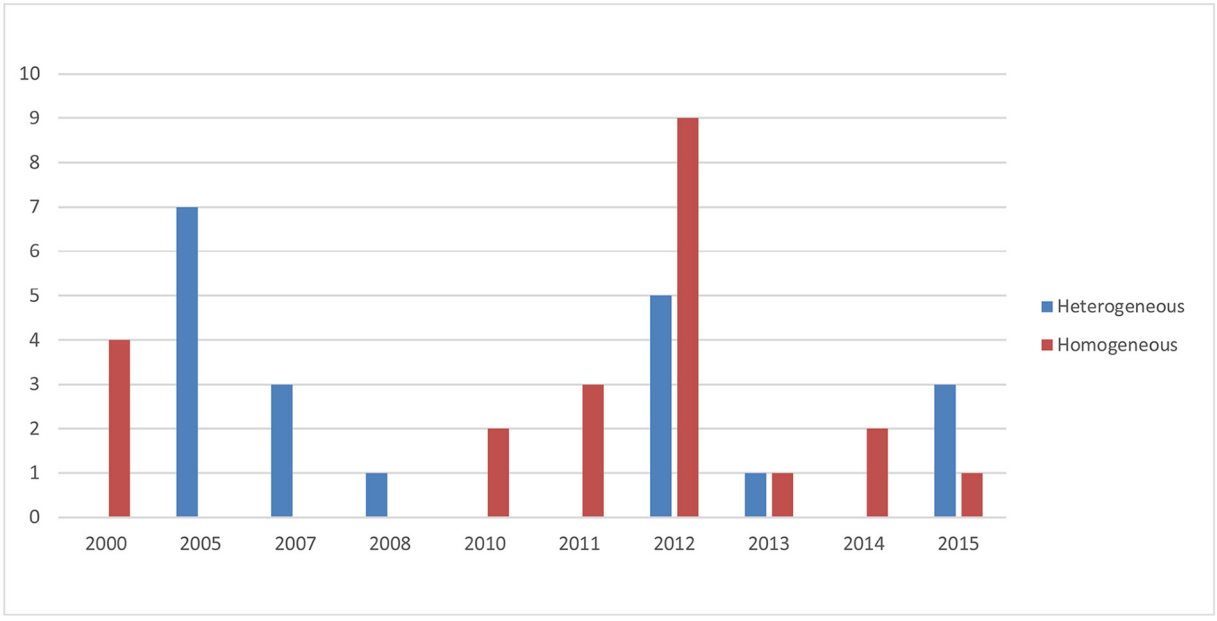

Fig. 1 – Homogeneous vs. Heterogeneous ensembles for IDSs.

graphically. This work's overview of related approaches revealed several such techniques, namely:

- Stacking
- Averaging
- Weighted averaging
- Belief measurement
- Dempster-Schafer combination
- Majority voting
- Weighted majority voting

Based on this analysis of heterogeneous ensembles in IDSs, as illustrated graphically in Fig. 2, it may be noted that multiple-classifier systems based on weighted majority voting are rarely implemented for this task. Therefore, it is one of the aims of this study to explore the benefits of WMV ensemble classifiers for classification of network traffic, as represented by the NSL-KDD data set.

As observed in section 4.3, there are many examples of WMV-based classification systems. However, this technique is rarely used in IDSs based on heterogeneous ensembles. The recommendation of this study is not only to develop a WMV heterogeneous ensemble for IDSs but also to devise a novel way of constructing such an ensemble. The above overview of popular approaches for WMV heterogeneous ensembles isolated two points of interest:

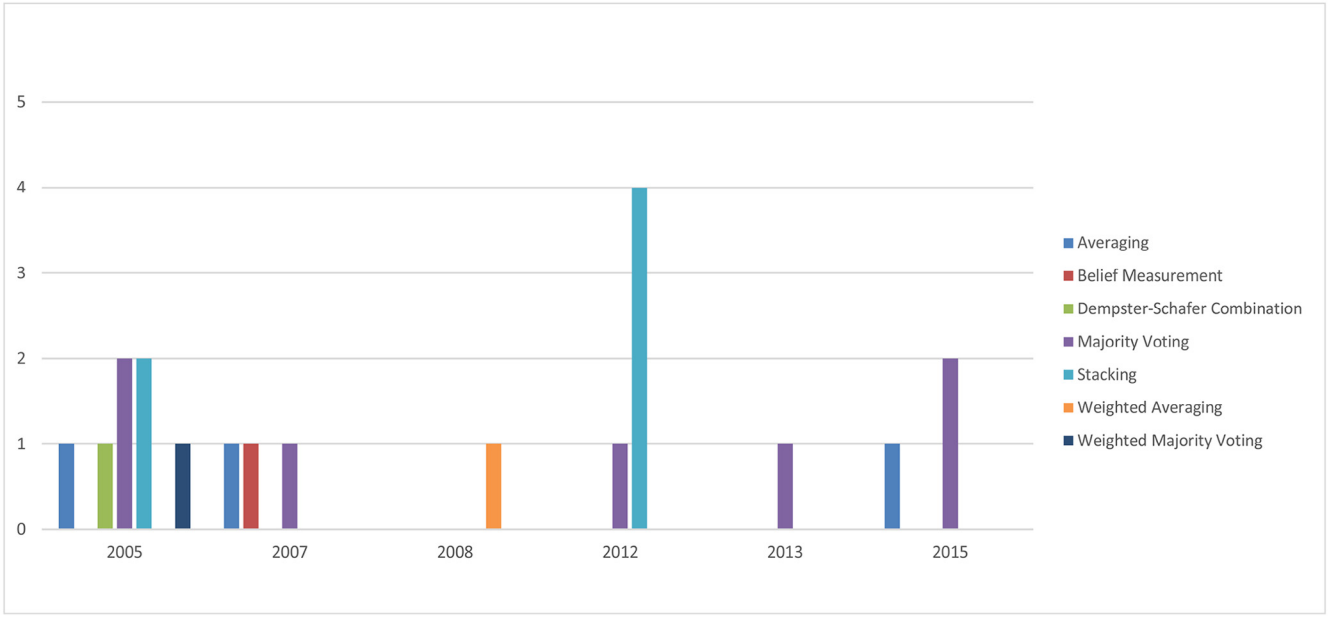

Fig. 2 – Types of heterogeneous ensembles for IDSs.

1. The weighting scheme, which defines how the reliability of each classifier is measured, and
2. The weight-generation method, which defines the values of weight coefficients used to measure the reliability of each classifier

The selection of an adequate weighting scheme, as its primary component, is a prior requirement for any proposed ensemble system. Although it is a theoretically sound concept, it may be noted that the class recall weighting scheme proposed by Kuncheva and Rodríguez (2014) is rarely used. With a class recall-based weighting scheme, the set of weights is needed for each classifier in the ensemble, where each weight in the set represents the reliability of the classifier for each class in the data set, which is also known as "class recall." Two other research studies (Zhang et al., 2011; Remya and Ramya, 2014), have also examined the viability of a class recall-based weighting scheme, with varying degrees of success. The general unpopularity of this approach may be due to difficulties in determining appropriate values for the weights. Therefore, another recommendation in this study is to develop a weight generation method that can successfully mitigate that problem.

Finally, regarding the various ways of determining optimal weight coefficients, it was noted that a meta-heuristic optimization approach deserves more attention. De Stefano et al. first demonstrated the effectiveness of the idea (De Stefano et al., 2002), as did Kausar et al. (2010). Therefore, utilization of meta-heuristic optimization to find the near-optimal sets of weight coefficients for a voting scheme based on class recall is a recommended way of generating heterogeneous ensemble classifiers.

REFERENCES

Angelov PP, Zhou X. Evolving fuzzy-rule-based classifiers from data streams. IEEE Trans Fuzzy Syst 2008;16(6):1462–75.

Axelsson S. Intrusion detection systems: a survey and taxonomy, Tech. rep., Technical report Chalmers University of Technology, Goteborg, Sweden; 2000.

Bahri E, Harbi N, Huu HN. Approach based ensemble methods for better and faster intrusion detection. In: Computational intelligence in security for information systems. Springer; 2011. p. 17–24.

Borji A. Advances in computer science – ASIAN 2007. Berlin, Heidelberg: Springer Berlin Heidelberg; 2007. p. 254–60 Ch. Combining Heterogeneous Classifiers for Network Intrusion Detection.

Breiman L. Bagging predictors. Mach Learn 1996;24(2):123–40.

Breiman L. Pasting small votes for classification in large databases and on-line. Mach Learn 1999;36(1–2):85–103.

Breiman L. Random forests. Mach Learn 2001;45(1):5–32.

Bukhtoyarov V, Zhukov V. Ensemble-distributed approach in classification problem solution for intrusion detection systems. In: Intelligent data engineering and automated learning–IDEAL 2014. Springer; 2014. p. 255–65.

Chan APF, Ng WWY, Yeung DS, Tsang ECC. Comparison of different fusion approaches for network intrusion detection using ensemble of RBFNN. In: 2005 international conference on machine learning and cybernetics, vol. 6. 2005. p. 3846–51 doi:10.1109/ICMLC.2005.1527610.

Chawla NV, Hall LO, Bowyer KW, Moore T Jr, Kegelmeyer WP. Distributed pasting of small votes. In: International workshop on multiple classifier systems. Springer; 2002. p. 52–61.

Chen Y, Zhao Y. A novel ensemble of classifiers for microarray data classification. Appl Soft Comput 2008;8(4):1664–9.

Chen Y, Wong M-L, Li H. Applying ant colony optimization to configuring stacking ensembles for data mining. Exp Syst Appl 2014;41(6):2688–702.

Cheng J, Chen L. A weighted regional voting based ensemble of multiple classifiers for face recognition. In: International symposium on visual computing. Springer; 2014. p. 482–91.

Cheng W, Hüllermeier E. Combining instance-based learning and logistic regression for multilabel classification. Mach Learn 2009;76(2–3):211–25.

De la Hoz E, de la Hoz E, Ortiz A, Ortega J, Martínez-Álvarez A. Feature selection by multi-objective optimisation: application to network anomaly detection by hierarchical self-organising maps. Knowl Based Syst 2014;71:322–38.

De Stefano C, Cioppa AD, Marcelli A. An adaptive weighted majority vote rule for combining multiple classifiers. In: Proceedings. 16th international conference on pattern recognition, 2002, vol. 2. 2002. p. 192–5 doi:10.1109/ICPR.2002.1048270.

Dietterich TG. Ensemble methods in machine learning. In: Multiple classifier systems. Springer; 2000. p. 1–15.

Eid HF, Darwish A, Hassanien AE, Kim T. Intelligent hybrid anomaly network intrusion detection system. In: Communication and networking. Springer; 2011. p. 209–18.

Eleyan A, Özkaramanli H, Demirel H. Weighted majority voting for face recognition from low resolution video sequences. In: Computing with words and perceptions in system analysis, decision and control, 2009. ICSCCW 2009. Fifth international conference on soft computing. IEEE; 2009. p. 1–4.

Enache AC, Patriciu VV. Intrusions detection based on support vector machine optimized with swarm intelligence. In: 2014 IEEE 9th international symposium on applied computational intelligence and informatics (SACI). 2014. p. 153–8 doi:10.1109/SACI.2014.6840052.

Ferri C, Flach P, Hernández-Orallo J. Delegating classifiers. In: Proceedings of the twenty-first international conference on machine learning. ACM; 2004. p. 37.

Folino G, Pizzuti C, Spezzano G. An ensemble-based evolutionary framework for coping with distributed intrusion detection. Genet Program Evolvable Mach 2010;11(2):131–46.

Freund S. A desicion-theoretic generalization of on-line learning and an application to boosting. In: European conference on computational learning theory. Springer; 1995. p. 23–37.

Fürnkranz J, Hüllermeier E, Mencía EL, Brinker K. Multilabel classification via calibrated label ranking. Mach Learn 2008;73(2):133–53.

Gaikwad D, Thool RC. Intrusion detection system using bagging with partial decision treebase classifier. Procedia Comput Sci 2015;49:92–8.

Gama J, Brazdil P. Cascade generalization. Mach Learn 2000;41(3):315–43.

Govindarajan M, Chandrasekaran R. Intrusion detection using an ensemble of classification methods. In: World congress on engineering and computer science, vol. 1. 2012. p. 1–6.

Gu S, Jin Y. Heterogeneous classifier ensembles for EEG-based motor imaginary detection. In: 2012 12th UK workshop on computational intelligence (UKCI). IEEE; 2012. p. 1–8.

Gu Y, Zhou B, Zhao J. PCA-ICA ensembled intrusion detection system by pareto-optimal optimization. Inform Technol J 2008;7:510–15.

Gudadhe M, Prasad P, Wankhade K. A new data mining based network intrusion detection model. In: 2010 international conference on computer and communication technology (ICCCT). IEEE; 2010. p. 731–5.

Hansen LK, Salamon P. Neural network ensembles. IEEE Trans Pattern Anal Mach Intell 1990;12:993–1001.
Haq NF, Onik AR, Shah FM. An ensemble framework of anomaly detection using hybridized feature selection approach (HFSA). In: SAI intelligent systems conference (IntelliSys), 2015. 2015. p. 989–95 doi:10.1109/IntelliSys.2015.7361264.
Hassanien AE, Kim T-H, Kacprzyk J, Awad AI. Bio-inspiring cyber security and cloud services: trends and innovations, vol. 70. Springer; 2014.
Hota H, Shrivas AK. Data mining approach for developing various models based on types of attack and feature selection as intrusion detection systems (IDS). In: Intelligent computing, networking, and informatics. Springer; 2014. p. 845–51.
Huang YS, Suen CY. The behavior–knowledge space method for combination of multiple classifiers. In: IEEE computer society conference on computer vision and pattern recognition. Institute of Electrical Engineers Inc (IEEE); 1993. p. 347.
Jacobs RA, Jordan MI, Nowlan SJ, Hinton GE. Adaptive mixtures of local experts. Neural Comput 1991;3(1):79–87.
Jankowski N, Grabczewski K. Heterogenous committees with competence analysis. In: Fifth international conference on hybrid intelligent systems, 2005. HIS'05. IEEE; 2005. p. 6.
Jordan MI, Jacobs RA. Hierarchical mixtures of experts and the EM algorithm. Neural Comput 1994;6(2):181–214.
Jordan MI, Xu L. Convergence results for the EM approach to mixtures of experts architectures. Neural Netw 1995;8(9):1409–31.
Kanakarajan NK, Muniasamy K. Improving the accuracy of intrusion detection using GAR-Forest with feature selection. In: Proceedings of the 4th international conference on frontiers in intelligent computing: theory and applications (FICTA) 2015. Springer; 2016. p. 539–47.
Kausar A, Ishtiaq M, Jaffar MA, Mirza AM. Optimization of ensemble based decision using PSO. In: Proceedings of the world congress on engineering, WCE, vol. 10. 2010. p. 1–6.
Kumar G, Kumar K. Design of an evolutionary approach for intrusion detection. Scientific World Journal 2013;2013:962185.
Kuncheva LI, Rodríguez JJ. A weighted voting framework for classifiers ensembles. Knowl Inf Syst 2014;38(2):259–75.
Lee W, Stolfo SJ, Mok KW. Adaptive intrusion detection: a data mining approach. Artif Intell Rev 2000;14(6):533–67.
Lin L, Zuo R, Yang S, Zhang Z. SVM ensemble for anomaly detection based on rotation forest. In: 2012 third international conference on intelligent control and information processing (ICICIP). IEEE; 2012. p. 150–3.
Malik AJ, Shahzad W, Khan FA. Binary PSO and random forests algorithm for probe attacks detection in a network. In: 2011 IEEE congress on evolutionary computation (CEC). IEEE; 2011. p. 662–8.
Masarat S, Taheri H, Sharifian S. A novel framework, based on fuzzy ensemble of classifiers for intrusion detection systems. In: 2014 4th international conference on computer and knowledge engineering (ICCKE),. IEEE; 2014. p. 165–70.
Meng Y, Kwok L-F. Enhancing false alarm reduction using voted ensemble selection in intrusion detection. Int J Comput Intell Syst 2013;6(4):626–38.
Mikolov T., Chen K., Corrado G., Dean J. Efficient estimation of word representations in vector space, arXiv preprint arXiv:1301.3781.
Miranda Dos Santos E. Static and dynamic overproduction and selection of classifier ensembles with genetic algorithms. Canada: Ecole de Technologie Superieure; 2008.
Mukkamala S, Sung AH, Abraham A. Intrusion detection using an ensemble of intelligent paradigms. J Netw Comput Appl 2005;28(2):167–82.
Pajouh HH, Dastghaibyfard G, Hashemi S. Two-tier network anomaly detection model: a machine learning approach. J Intell Inf Syst 2015;1–14.
Pervez MS, Farid DM. Feature selection and intrusion classification in NSL-KDD cup 99 dataset employing SVMS. In: 2014 8th international conference on software, knowledge, information management and applications (SKIMA),. 2014. p. 1–6 doi:10.1109/SKIMA.2014.7083539.
Pfahringer B. Winning the kdd99 classification cup: bagged boosting. SIGKDD Explor 2000;1(2):65–6.
Rastegari S, Hingston P, Lam C-P. Evolving statistical rulesets for network intrusion detection. Appl Soft Comput 2015;33:348–59.
Read J, Pfahringer B, Holmes G, Frank E. Classifier chains for multi-label classification. Mach Learn 2011;85(3):333–59.
Remya K, Ramya J. Using weighted majority voting classifier combination for relation classification in biomedical texts. In: 2014 international conference on control, instrumentation, communication and computational technologies (ICCICCT). IEEE; 2014. p. 1205–9.
Richiardi J, Drygajlo A. Reliability-based voting schemes using modality-independent features in multi-classifier biometric authentication. In: Multiple classifier systems. Springer; 2007. p. 377–86.
Rogova G. Combining the results of several neural network classifiers. Neural Netw 1994;7(5):777–81.
Schapire RE. The strength of weak learnability. Mach Learn 1990;5(2):197–227.
Shi L, Xi L, Ma X, Weng M, Hu X. A novel ensemble algorithm for biomedical classification based on ant colony optimization. Appl Soft Comput 2011;11(8):5674–83.
Singh R, Kumar H, Singla R. An intrusion detection system using network traffic profiling and online sequential extreme learning machine. Exp Syst Appl 2015;42(22):8609–24.
Syarif I, Zaluska E, Prugel-Bennett A, Wills G. Application of bagging, boosting and stacking to intrusion detection. In: Machine learning and data mining in pattern recognition. Springer; 2012. p. 593–602.
Tahir MA, Kittler J, Bouridane A. Multilabel classification using heterogeneous ensemble of multi-label classifiers. Pattern Recognit Lett 2012;33(5):513–23.
Tama BA, Rhee KH. A combination of pso-based feature selection and tree-based classifiers ensemble for intrusion detection systems. In: Advances in computer science and ubiquitous computing. Springer; 2015. p. 489–95.
Tavallaee M, Bagheri E, Lu W, Ghorbani A-A. A detailed analysis of the KDD cup 99 data set. In: Proceedings of the second IEEE symposium on computational intelligence for security and defence applications. 2009. 2009. p. 1–6.
Toman H, Kovacs L, Jonas A, Hajdu L, Hajdu A. Generalized weighted majority voting with an application to algorithms having spatial output. In: International conference on hybrid artificial intelligence systems. Springer; 2012. p. 56–67.
Tsoumakas G, Katakis I, Vlahavas I. Effective voting of heterogeneous classifiers. In: European conference on machine learning. Springer; 2004. p. 465–76.
Tsoumakas G, Katakis I, Vlahavas I. Mining multi-label data. In: Data mining and knowledge discovery handbook. Springer; 2009. p. 667–85.
Valdovinos RM, Sánchez JS. Combining multiple classifiers with dynamic weighted voting. In: International conference on hybrid artificial intelligence systems. Springer; 2009. p. 510–16.
Van Erp M, Schomaker L. Variants of the Borda count method for combining ranked classifier hypotheses. In: In the seventh international workshop on frontiers in handwriting recognition. 2000. Amsterdam learning methodology inspired by human's intelligence Bo Zhang, Dayong Ding, and Ling Zhang. Citeseer; 2000. p. 443–52.
Wolpert DH. Stacked generalization. Neural Netw 1992;5(2):241–59.

Ye F, Zhang Z, Chakrabarty K, Gu X. Board-level functional fault diagnosis using artificial neural networks, support-vector machines, and weighted-majority voting. IEEE Trans Comput Aided Des Integ Circ Syst 2013;32(5):723–36.
Zhang M-L, Zhou Z-H. ML-KNN: a lazy learning approach to multi-label learning. Pattern Recognit 2007;40(7): 2038–48.
Zhang X, Wang P, Du L, Liu H. New method for radar HRRP recognition and rejection based on weighted majority voting combination of multiple classifiers. In: 2011 IEEE international conference on signal processing, communications and computing (ICSPCC). IEEE; 2011. p. 1–4.

Abdulla Amin Aburomman is a Ph.D. candidate in National University of Malaysia, in the Department of Electrical, Electronic & Systems Engineering, Faculty of Engineering and Built Environment since February 2012. His research interests include: Machine Learning, Optimization, Data Mining, and Pattern Recognition. He has been involved in several conferences and workshops dealing with Machine Learning and Optimization. He has published several academic works with large publishers, i.e. Elsevier. He has done several peer reviews for different high impact factor Journals in his area of expertise.

Mamun Bin Ibne Reaz is a Senior Member of IEEE and currently a Professor in the Department of Electrical, Electronic and Systems Engineering, National University of Malaysia. He is involved in teaching, research and industrial consultation. Dr. Reaz has vast research experience in Japan, Italy and Malaysia. He is the author and co-author of 200+ research articles in the field of design automation and IC design for biomedical applications. He is also the recipient of more than 50 research grants (national and international). He received his D.Eng. degree in 2007 from Ibaraki University, Japan. Vlsi dDesign, Biomedical Application Ic.