Classification and Clustering Based Ensemble Techniques for Intrusion Detection Systems: A Survey

Nabeel H. Al-`araji 1, Safaa O. Al-mamory 2, Ali H. Al-Shakarchi 3

1 Ministry of Higher Education, Baghdad, Iraq
2 University of Information Technology and Communications, Collage of Business Informatics, Baghdad, Iraq.
3 University of Babylon, Collage of Information Technology, Hillah, Iraq.

E-mail: ali.alshakarchi@itnet.uobabylon.edu.iq

Abstract. A huge amount of data is transmitted through the networks, which allowed the exchange of knowledge and medical expertise, trade and banking facilities, etc. However, due to the huge connections to these networks, the security issue has been floated on the surface. Intrusion Detection System (IDS) plays a significant role to protect computer systems. To compensate these issues, the orientation is to employed machine learning and data mining techniques to design and implement powerful IDSs. Among these techniques is ensemble learning which enables a combination of multiple models to enhance overall performance. This study presents a brief overview of IDSs, discusses the history of ensemble systems, specifies the methods adapted in designed such system, highlights the most important ensemble techniques, demonstrates in detail the main methods that have been adapted in combining ensemble components. Besides, special attention was paid to studies in the period (2009-2020) that focus onto both ensemble classification and clustering when developing IDSs.

1. Introduction
Nowadays, network security has become the most crucial thing that gives rise to concern for corporations and organizations. More and more attacks are executed in a very professional way in both cooperate or private networks. New attacks are emerging day after day and lots of previous attacks remain current. As these attacks success performed in the target networks, it cases very harmful consequences in both financial and economic aspects [1,2]. For example, according to the Internet Security Threat Report (ISTR) [3], there are 4,818 unique websites were infected with formjacking1 malicious attack every month in 2018 that yield loss of up to 2.2 million dollars per month.

Although there are many network protection methods are existing like access control, encryption, authentication, and advanced firewall, there is an urgent need to intelligent Intrusion Detection System (IDS) that can detect known and a novel attack automatically [4,5].

1 Formjacking is an attack that uses malignant Java code to larceney credit cards and other sensitive information from people who shop online from commercial sites.
2. Intrusion Detection Systems (IDSs)
Ni, He, Chan and Ahmad [6] defined intrusion behavior as follows: "any set of actions that attempt to compromise the integrity, confidentiality or availability of a resource". For mitigating this malicious behavior, Intrusion Detection Systems (IDSs) are widely adopted in the worldwide to make an impenetrable barrier as possible against the intruders [7]. The main task of such a system (i.e. IDS) is to monitor and identify abnormal (malicious) activity in computing devices or networks and send an alarm to the administrator. So, it is a major challenge for any IDSs to classify the incoming pattern to determine if it is normal or not with the highest accuracy and lowest false positive as possible [8].

2.1. Host-Based VS Network-Based IDS
According to the information source, IDSs could be grouped into Host-based and Network-based. Host-based IDS (HIDS) in particular focuses on host data like identifiers and system calls during detecting intrusion behaviors. Thus, protecting a single host but the network as a whole is vulnerable to attacks. While the Network-based IDS (NIDS) major concerns are on network-related data like traffic volume, source/destination IP address, protocol usage, port number, etc., and consequently protect the hole network [9].

2.2. Misuse Detection VS. Anomaly Detection IDS
Two major detection techniques are adapted when designing an IDSs: Misuse and Anomaly detection techniques. Misuse IDS (MIDS), sometimes called signature-based detection, counting on attack signatures (i.e. previously known attacks) to detect malicious attacks. Whenever the patterns under inception and the signatures of known attack are identical, an alert is released and send to the security administrator. This type of IDSs characterized by the high speed of detection with low false positive alarm. However, it fails to detect new emerging attacks (also known as zero-day attacks) and this leads to that MIDS continuously required expert inception to identify new signature which is expensive and make the system for a period of time under vulnerabilities till discovering the signatures for new attacks [10].

The well-known Snort tool [11] operates as MIDS. On the other hand, Anomaly IDS (AIDS) has adopted a different approach to detect intrusion. Instead of creating a signature for every attack and save them into a database, it identifies the behavioral pattern and discovers if it does not conform to expected behavior. More precisely, AIDS relays on storing the usual behaviors (i.e. training on normal activities) then comparing the current one with the trained behaviors. If there is a deviation huge enough between trained and current behavior, we can say that smoothing abnormal is occurred [12]. Although AIDS can detect zero-day attacks, it suffering from a high false-positive rate [9].

2.3. Rule VS. Model VS. Statistical IDS-Based System
Shu Yun Lim and Andy Jones [13] provide classification regarding AIDS: they are rule, model, and statistical based IDS. The first taxonomy (i.e. Rule-based IDS) characterized the normal behavior of the users, networks, and computer systems via a set of rules. Expert systems lie in this category. The second taxonomy is model-based which is the opposite of rule-based, try to model the intrusions at a high level of abstraction. In these systems, anomalies are detected as deviations from the model that represents normal behavior. Machine learning and data mining are a vital example of this category. In the statistical-based IDS systems, the anomaly detector observes an object and build a profile to represent its stochastic behavior. Thereafter compare the obtained profile with previous predefined profile to detect anomalies. Statistical-based need not previous knowledge about the normal behavior and provide accurate alarm occurring over a long period. These techniques susceptible to being trained by an attacker, and require a long time to report the anomaly for the first time [14].
2.4. *Supervised vs. Unsupervised vs. Semi-Supervised learning*

According to the learning method, IDSs could be trained by utilizing three main types of machine learning (ML)/data mining (DM) approaches: supervised, unsupervised, and semi-supervised [15]. In supervised learning, which sometimes called (direct classification or predictive), a set of training data is given in advance, and employ one of a supervised algorithm for prediction. A large spectrum of a well-known supervised learning algorithm is available like Support Vector Machine (SVM), Artificial Neural Network (ANN), Logistic Regression (LR), Naïve Bayes (NB), K-Nearest Neighbor (KNN), Random Forest (RF), Decision Trees (DT), and etc. The unsupervised learning, referred to as (descriptive or undirected classification), there's no training data that's data without the specified output. Examples of such learning are clustering, Self-Organizing Map (SOM), etc. Semi-supervised is considered as a hybrid learning technique that required both labeled and unlabeled data to derive the model [16]. IDSs categories could be summarized in Table 1.

| IDS characteristics | Type                        | Description                                                                 | Pros/Cons (notes)                                                                 |
|---------------------|-----------------------------|----------------------------------------------------------------------------|---------------------------------------------------------------------------------|
| According to        | Host-Based                  | Mainly focus on host data such as identifiers and system calls              | Protect single host/ The network as a whole is vulnerable to attacks             |
| information source  | Network-Based               | Focus on network-related data such as traffic volume, source/destination IP address, protocol usage, and port number, etc. | Protect the hole network                                                        |
| According to        | Misuse detection like SNORT [11] | Detect known attacks. Once network data is matched with attack signatures, an alarm is generated. | High detection rate, low false positive/ fail to detect a novel attack, require expert inception to identify new signature(expensive) |
| the analysis        | Anomaly detection           | Detect unknown attack by identifying deviations from normal activities      | Detect novel attack/ high false positive                                          |
| procedure           | Statistical-Based           | Use statistical methods to detect an attack, they monitor the traffic for a long time and report an alarm if any abrupt change. | Do not require prior knowledge of the normal activity, provide accurate alarm occurring over a long period/ susceptible to being trained by an attacker, and require a long time to report the anomaly for the first time. |
| According to        | Data mining and machine     | Aim to discover patterns from a dataset and build a model to detect attacks. | Can detect a known and unknown attack, the ability to process a large amount of data, discover hidden information in the data. |
| detection methods   | learning-Based              |                                                                             |                                                                                  |
|                     | Supervised                  | Known as (predictive or direct classification). A set of training data is given in advanced, and employ one of a supervised algorithm for prediction. | simple to understand, keep just decision boundary in memory after training/ training need a lot of computation time, need prior knowledge about classes in the data |

Table 1. Intrusion detection systems categories
There is no training data that is data without the desired output. Work without prior knowledge about the classes, can cluster or classify data by discovering their feature/ more complicated than supervised.

Hybrid method required a small sample of labeled data for learning task Work fine when a small subset of labeled data is available

3. Ensemble Based Systems

In most important and grates matters like medical, financial, social, security, or any other inclusions, we often try to find out a second or maybe third or sometimes more opinion before making our final decision. In this process, we give avoirdupois the individual opinions and eventually combine them to get the result that supposed to be the most reliable one. In particular, the system that depends on consulting several experts before making a final judgment called Ensemble System[17].

Ensemble is a charming Machine Learning (ML) technique that combines two or more classifiers' opinions to make the final decision, which is often better than individuals. This technique is more flexible to noise compared with a single classifier. Besides, it uses a divide and conquers concept in which an intricate task is fragmented into several sub-tasks that are easier to handle and resolve[18].

The common ensemble architecture could be viewed in figure 1.

![Figure 1: A Common ensemble architecture](image)

3.1. Brief History of Ensemble Systems

Determining the exact historical starting point for ensemble methods is very complicated due to the fact that the conception of implementing several models has been used in the human community previously. But in the last decades, researchers in the fields like pattern recognition, machine learning, data mining, and statistics have utilized ensemble methods in different aspects [19].

To the best of our knowledge, Dasarathy et al [20] present their first paper on the ensemble concept in 1979 when they suggest partitioning the feature space using two or more classifiers. Later in1990 Hansen et al [21] go farther and find out that predictions made via a combination of classifiers are often more accurate than the prediction made by a single classifier. Other interesting theory around the ensemble presented by Schapire [22] also in 1990 who’s put the cornerstone of ensemble system in the machine learning area, which confirmed that feeble learners could be elevated to robust learners. All previously mentioned works open promising orientation toward utilizing and suggesting new ensemble methods under different names and ideas. For example, composite classifier systems [20], a mixture of experts [23], stacked generalization [24], a combination of multiple classifiers [25], and many others.
3.2. Ensemble Methods
For constructing ensemble of classifiers, the first step is to prepare a number of learners. These learners (which also called base learner) are constructed from training data via a learning algorithm that could be of any type like Decision Tree (DT), Neural Network (NN), Support Vector Machine (SVM), or any other type from large spectrum of learning algorithms. After training phase is completed, each base learner has acquired the ability to classify unseen incoming instance. Thereafter, in the testing phase, each base learner gives its own opinion for new input, and the last decision is made up via combining their outputs [19]. The next sub-section will reveal ensemble methods according to different point of view.

3.2.1. Ensemble structure
Two major ensembles could be distinguished based on ensemble's structure. They are homogenous and heterogeneous ensemble [14]. In the former, an ensemble is made up from two or more learners of the same type. While the later utilizes learners of the different types. In particular, homogenous ensemble could be viewed as a simple structure for creating several hypotheses which are extended from the same single classification algorithm via constructing several variations of that classifier. The basic idea of homogenous structure is that, aggregate several relatively simple decisions will eventually lead to more reliable decision [26,27]. The later (i.e. heterogeneous), an ensemble decision relies on the aggregation of diverse base learner decisions. More precisely, the term heterogeneous refers to implication of different classification techniques to make a decision process. The attractive point of such structure is that, each base learner has been created based on different learning algorithm which is mean that each one has its own and different capability for classification task. The final output of each learner must be aggregating and combined to obtain the final result [19,28]. Almost all ensemble combination strategies will be covered in this survey.

3.2.2. Ensemble construction (Ensemble diversity)
Robi Polikar [17] referred to two important questions that must be taken into account when designing an ensemble system. First, how the base learners will be created? and second how they differ from each other? The answer to these questions eventually will determine the degree of ensemble diversity which is very important criteria for improving the ensemble system.

In the same context, Hansen and Salamon [21] believed that for constructing an accurate ensemble system which is better than its components, the constitute classifiers should be accurate and diverse. It could be said that two classifiers are diverse if they made an uncorrelated error on new input instance. To explain the idea behind the importance of diversity concept, let consider the following classification scenario: suppose there exist an ensemble with three classification models (K1, K2, K3) and suppose new input instance x. If all ensemble constituent classifiers are conformable (i.e. they are not diverse), then as one classifier supposes K1(x) is erroneous, this led that K2(x) and K3(x) will also be erroneous. But consider that the error of these classifiers is uncorrelated, then if K1(x) is erroneous, K2(x) and K3(x) may be correct. Thus, combining these three classifiers via majority vote will eventually classify input x correctly [29]. From the previous scenario, it clear that diversity in the classifier's outputs is vital and the first key for constructing accurate ensemble system. More precisely, we need that ensemble constituent classifiers to be as true as possible, and if there are some failures in the classification process, these failures should be on different instances[30]. However, [17,18] clarified that there are four main ways to reach diversity criteria which is also could be considered as a method of ensemble construction: (i) manipulating training data (ii) manipulating training parameters (iii) manipulating input features (iv) manipulating the class labels (v) manipulating learning algorithm.

(i) Manipulating Training Data
In this method, a diverse training sets are employed for training the ensemble members. These training set usually gained via resampling techniques according to some sampling distribution where the new
sampled dataset is gotten randomly with replacement from the original dataset. Bagging as well as boosting are a vital example of such approaches to achieve diversity criteria [31,32].

(ii) Manipulating The Input Features
Diversity could also have obtained by using a different subset of input features to train individual classifiers. These input features are chosen either stochastic or upon on the validation of experts. It's worth to mention that constructing base learners utilizing arbitrary feature subsets is branded as a random subspace scheme which is first introduced by [33]. Random Forest uses manipulating input features and utilizes DTs in place of its base learners [32].

(iii) Manipulating Training Parameters
Alternative method to reach diversity is to use dissimilar training parameters for each classifier. As instance: suppose that we have an ensemble which is a composite of a number of Multi-Layer Perceptron (MLP) as base learners. Manipulating initialization weights, number of layers, number of nodes, etc., will eventually allow one to adjust the instability of the ensemble components, and hence control their diversity [31].

(iv) Manipulating The Class Labels (Error-Correcting Output Coding (ECOC) Method)
Diversity also could be enforced by manipulating classifier outputs. This means that each classifier is oriented to classify just some classes in the dataset. This method is originally employed whenever the classes in the dataset are relatively large (i.e. multi-class problem). The core of the method is to transform training data into two-class problem via separating classes into two disjoint subsets, like Z0 and Z1. Then, assigning training instances whose class refers to Z0 into class 0 while others to class 1. These relabeled instances are utilized toward training a base learner. By recurring this learning (i.e. class relabeling) several times, an ensemble with diversity criteria is obtained. In the testing phase, all classifiers are given their prediction, then if the predicted instance refers to class 0, all the classes belong to Z0 have been received a vote and vice versa. At first, votes are calculated and the class which get a larger vote is bended to the test instance [27,32]. One vital example of such a method is the “Error-Correcting Output Coding” method which is proposed by Dieterich and Bakiri [34].

(v) Manipulating The Learning Algorithm
Constructing an ensemble system with different base learner algorithms will lead to diversity due to the biases of ensemble components. For example, using a different type of classifier, like Support Vector Machine (SVM), decision trees, nearest neighbor, and Neural Network (NN) to construct ensemble system can introduce diversity to the system (i.e. heterogeneous classifiers). However, the combination method should be chosen carefully when combining different model types or maybe a dissimilar structural design of the same model [17,27]. Ylias Chali et al derive an attractive question as to the title for their paper “Complex Question Answering: Homogeneous or Heterogeneous, Which Ensemble Is Better?”. They eventually conclude that a heterogeneous ensemble system for solving complex tasks is desirable [35]. For the previous consideration, our proposed ensemble systems (EBREMs & SBRC) will utilize heterogeneous base classifiers (NB & J48) to ensure diversity in the produced systems.

3.2.3. Ensemble Techniques
For constructing an ensemble of learners, three main techniques could be used for this purpose named Bagging, Boosting, and Stacking [36,37].
A. Bagging

The term bagging is an acronym for “Bootstrap AGGregatING” was first introduced by Breiman [38]. The core idea of bagging is simple and straightforward: an ensemble is made up of classifiers which are often of the same type trained on generated random bootstrap training subset, of course with replacement; from the original training dataset. After the training phase, the classifier outputs are combined and the final result is obtained by voting [31,39]. Figure 2 illustrates the bagging method.

![Figure 2: Illustration of bagging method.](image)

The success achieved by bagging is due two reasons: first, the newly generated bootstrap training subsets produce results that are characterized statistically with a minimum variant, and second, is because the combination of decisions made by ensemble constituent classifiers is more tolerant to outlier’s error[40]. The algorithm of bagging is summarized in figure 3 [41].

**Algorithm: Bagging**

Input:
- D a set of d training tuples.
- k the number of models in the ensemble.
- A classification learning scheme.

Output: Bagging Ensemble Model.

1. Begin
2. for i = 1 to k do // construct k models:
3. Create bootstrap sample Di by sampling D with replacement.
4. Use Di and the learning scheme to derive a model Mi.
5. end for
6. End

To use the ensemble model for classifying instance x: let each of the k models classify x and combine the results via majority vote.

![Figure 3: Bagging algorithm](image)

It remains to be mentioned that Bauer et al [42] proposed a variation of bagging called Wagging which is an acronym of (WeightAggregation). In wagging, weights generated in boosting instances are utilized to direct the selection of a bootstrap sample. Wagging adds noise to each weight with mean zero and specified standard deviation, then induce the learner. Eventually, instances will disappear from training when their weight decreased to zero.
B. Boosting

The gist behind boosting could be illustrated by the following example: Assume that a sick person has shown a set of confirmed symptoms. Rather than making a consultation for just a single doctor, one can seek to consult several doctors, assigning a value (i.e. weight) to each of their opinions based on the rigor of diagnoses they previously have done. Then, the ultimate decision is one obtained by combining of weighted diagnoses [41]. In machine learning, boosting, which is also called AdaBoost as an acronym of AdaptiveBoosting; is introduced early based on the question asked by Kearns [43]: “can a set of weak learners create a single strong learner?”. The weak learner is a classifier which is only slightly better than random guessing. In contrast, a strong learner is a classifier that considered to produce true classification [44]. More precisely, the boosting technique has tended to enhance the classification of a frail classifier via training it reiterated on training dataset taken from diverse distributions. Figure 4 will give brief imagination of boosting technique.

![Illustration of boosting technique.](image)

Freund and Schapire [45] introduce two versions of the AdaBoost (AdaBoostM1, and AdaBoostM2). The main differences between these variations and original AdaBoost are the way the ultimate hypothesis is obtained and how multi-class problems are addressed [46]. The boosting algorithm is illustrated in figure 5 [40].

**Algorithm: Boosting**

Input:
- D, distribution sample
- £, learning algorithm
- T, number of iterations

Output: Boosting Ensemble Model

1. Begin
2. \[ D1=D \] //create new distribution.
3. For i=1 to T do:
4. Train a base learner £ on Dt to obtain hi.
5. Measure ci the error of hi.
6. Determine new distribution Di+1 utilizing Di,ci.
7. end
8. End

![Boosting algorithm.](image)
C. Stacking
Stacking or sometimes called stacked generalization is an appealing ensemble technique that combines multiple classifiers to improve the classification result. Unlike bagging and boosting, stacking consist of two-level, the former level named base learner or called (level_0), while the second level is called (meta learner) or (level_1) [18]. As the first level uses heterogeneous classification models for learning from the training dataset, the output of these models will form a new dataset that serves as input to the stacking learner. In the new dataset (which contains a prediction of level_0 serve as an attribute) each instance will bind to the true class that should be predicted. The generated dataset is then utilized by meta learner to give the last finding [47]. Figure 6 shows the concept of stacking. Regarding the way that bagging boosting and stacking are trained, we say that bagging and stacking train their base learner in parallel fashion while AdaBoost training is done on sequential fashion. Stacking algorithm is summarized in figure 7 [19].

Algorithm: Stacking

Input:
- Data set D = {(x1, y1), (x2, y2),...,(xm, ym)}
- First-level learning algorithms L1,.., LT
- Second-level learning algorithm L
Output: Stacking Ensemble Model

1. Begin
2. for t= 1 to T do:
3.   ht = Lt(D);
4. end
5. D' = ∅;
6. for i= 1 to m do:
7.   for t= 1 to T do:
8.     zit = ht(xi);
9.   end
10. D' = D' ∪ ((zit1,..., ziT ), yi);
11. end
12. h' = L(D');
13. END

Figure 6: Illustration of Stacking

Figure 7: Stacking algorithm
3.2.4. Ensemble Combination Methods

The desired goal of an ensemble system is constructing a set of classification models (for classification task), and instead of trying to find the best one, ensemble techniques restore to combine these models for reaching extra precise classification result at the little cost of complication and with increased of generalization ability. [48] asked an attractive question in his book asking “whether a combination of classifiers is justified.”. He was found the satisfactory answer provided by [29] that explain in detail the benefit of ensemble combination and summarized it in three fundamental reasons:

i. Statistical Reason

Indeed, the ultimate goal of any machine learning algorithm is to find an optimal hypothesis h (i.e. learned model) in a given space H of hypotheses. This goal could be achieved if there is sufficient training dataset. However, in real cases, we only have just a limited training dataset. If learning algorithm, selects one of these hypotheses with good generalization performance, it is still a possibility that the selected hypothesis failed to forecast future data. It is possible to avoid this risk by averaging or combining these learned models, thus reducing the chance of choosing the wrong model as shown in figure 8(a).

ii. Computational Reason

Machine learning algorithms try to produce a learned model by employing search methods. In fact, these searching methods which perform local search may get fall in local optima trap. In this case, even if there is available sufficient training dataset, it may be still troublesome to catch out the superlative hypothesis. Alleviation of this troublesome could be reached by implementing the searching method via choosing different launching ways. Thus by combining hypotheses, we will decrease the chance of selecting a wrong local minimum as shown in figure 8(b).

iii. Representational Reason

In many machine learning tasks, it’s difficult to obtain the true hypothesis form the hypotheses in hypotheses space H. If we take the weighted sums of hypotheses in H, it is possible to expand the space of representable hypothesis as shown in figure 8(c).
Figure 8: Three major causes for ensemble combination (a) statistical, (b) computational, and (c) representational. $\mathcal{H}$ represent hypotheses space; $h$ represents individual hypothesis [29].

The second key to success any ensemble system relies on the strategies employed for combining constituent classifiers. Indeed, ensemble combinations could be grouped as trainable vs. non-trainable combination, or class label vs. class-specific continuous outputs combination. The next subsection will discuss these taxonomies.

A. Trainable Combination Rule
The trainable combination rule implies the training of the combiner module, either during or after the base learners have been trained [49]. The combiner of the ensemble in this combination strategy relays on a parameter, often called weight, which is determined via a separated training algorithm. Expectation Maximization (EM) utilized by the mixture-of-experts model is a vital instance of such combination rule. This strategy is considered as a dynamic combination rule since the parameter of rules are instance-specific [50].

B. Non-Trainable Combination Rule
The non-trainable combination does not require the training phase after the constituent classifiers have been induced (i.e. they apply ”fixed” rules to combine base classifiers) [49]. Examples of such combiners are weighted average and weighted majority vote, which is discussed later in this chapter. These combination rules are considered as non-trainable because parameters are available just after the model has been configured [48].
C. Class labels Combination

In this combination strategy, the class labels are only needed to accomplish the ensemble combination. To facilitate the combination methods of class labels, suppose the decision of the $t^{th}$ classifiers represented by $d_{t,j} \in \{0,1\}$, $t=1..T$ and $j=1..C$, where $T$ is a number of classifiers and $C$ is a number of classes. Here we say that if $t^{th}$ classifier choose class $w_j$, then $d_{t,j}=1$, and 0, otherwise.

1) Majority Vote

Majority Vote (MV) has three major versions:

- **Unanimous Voting**: a final decision required that all classifiers agree.
- **Simple Majority**: the final decision required that half plus one of classifiers in the pool are agreed.
- **Majority Voting (plurality voting)**: the final decision required a winning class should collect a large number of votes, regardless or not the sum of those votes exceeds 50%.

The decision made via majority vote can be illustrated in Equation (1) as follows: choose class $w_j$ if

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

The majority vote is considered as the most popular combination of class labels ensemble. Each model will give its a vote to specific class label, the ultimate decision represented by a class that receives at least half plus one of the votes [17,19]. The probability of the ensemble system to make a correct decision is proposed by Hansen and Salamon [21] which can be calculated based on the binomial distribution as the total probability of choosing $k \geq \lfloor T/2 \rfloor + 1$ successful one out $T$ of classifiers as illustrated in Equation (2).

$$P_{ens} = \sum_{k=\lfloor T/2 \rfloor + 1}^{T} \binom{T}{k} p^k (1-p)^{T-k}$$

Where $P_{ens}$ is the probability of ensemble success, $p$ the success rate of each classifier.

2) Weighted Majority Vote

Weighted Majority Vote (WMV) is an extension of the majority vote technique in which classifiers are assigned to different weights based on their performance. These weights of each model may be independent of the predicted class [38]. If there is proof that several models are more potent than others, then more weight is granted to decisions of them which may further enhance the general performance of the ensemble system. Let us refer the decision of hypothesis $h_t$ on class $w_j$ as $d_{t,j}$ such that $d_{t,j}=1$ if $h_t$ select $w_j$ and 0 otherwise. Also, suppose that we assign a weight $w_t$ to each model $h_t$ based on its performance. According to previous notation, ensemble system whose constitute classifiers are combined using WMV is select class $J$ according to Equation (3) below, if $\lfloor \cdot \rfloor$.

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

3) Behavior Knowledge Space (BKS)

A behavior knowledge space (BKS) originally introduced by Huang and Suen [51] is a $k$ dimensional space in which each dimension refers to the decision of one model. This method uses a table that created based on the classification of a training dataset which holds how often each combination of the label is produced by constitute classifiers [52]. Thus, the correct class that a specific labeling combination is seen frequently in the training phase will be chosen every time that combination occurs in the testing phase. BKS is shown in more details in figure 9. For demonstrating the procedure of BKS let us consider the combination $\{\omega_1 \omega_2 \omega_1\}$ occurs a total of 28 times, 10 of the combinations in which the correct class is $\omega_1$, 15 of them the correct class is $\omega_2$, and 3 of them the correct class is $\omega_3$. 
In this case, the winner class is $\omega_2$ since it's the foremost frequently observed true class for this mix. Thus, during the test phase, whenever the combination $(\omega_1 \omega_2 \omega_1)$ appears, the ensemble will choose $\omega_2$ as winning class [17].

![Figure 9: Behavior Knowledge Space Illustration[17]](image)

4) Borda Count

Borda count is employed when the classifiers have the ability to rank the classes. Each class receives a vote, the votes have been summed up through overall classifiers, and the final ensemble decision is the class label which receives the most votes. If we define $B_i(j)$ as the number of class labels ranked below class $\omega_j$ by classifier $i$, then the Borda count for class $\omega_j$ is $B_j$ defined in Equation (4) below [38]:

$$B_j = \sum_{i=1}^{t} B_i(j)$$  \hspace{1cm} (4)

Each instance under test is bounded to the class label with the largest Borda Count. It is worth noting that this combiner requires no training.

D. Continuous Outputs Combination

In the class label combination methods, each constitutes classifiers is produce output which is $\in \{0,1\}$ in the case of a binary classification task. In contrast, continuous output combination, the output of each classifier is treated as the grade of support that have been given to that class, moreover, it is generally interpreted as guesstimate of posterior probability for that class. The posterior probability needs to access adequate training data and the output is normalized to sum up to 1 overall class labels [27].

1) Algebraic Combination

Kuncheva, Bezdek and Duin [53] define a matrix called Decision Profile (DP) that helps to formulate algebra combination rules from a uniformed perspective. DP is considered a robust combination scheme that combines classifier outputs by comparing them to a sophisticated profile for each class. An attractive feather of DP is that; it combines all classifier outputs to calculate the ultimate support for each class. Statistically, the decision profile matrix $DP(x)$ for pattern $x$, having a range of elements $d_{t,j} \in [0,1]$ that represent the support given by $t$th classifier to class $w_j$. Rows belongs to $DP(x)$ characterize the support given by distinct constitute classifiers to each of the classes,
while the columns specify the support giver by specific class from all constitute classifiers. DP matrix is illustrated in figure 10 [17].

**Figure 10:** Decision Profile for Instance \( x \) [17]

Based on DP, it could define several combination rules as follows:

i. **Mean Rule**

It is possible to combine constitute classifiers via mean rule as follows: The confidence for class \( \omega_j \) is calculated by taking the mean of all classifiers’ \( j \) th outputs as in Equation (5).

\[
\mu_j(x) = \frac{1}{T} \sum_{t=1}^{T} d_{tj}(x) 
\]

The mean and sum rule is comparable, which is discussed later, except it has a normalization factor (i.e. \( 1/T \)). The ultimate output of ensemble is the class label \( w_j \) which formed with the maximum support of \( \mu_j(x) \) [36,40].

ii. **Weighted Average Rule**

It is a hybrid combination rule that extended form mean rule as well as WMV. In this combination rule, the weights have not applied to class labels, instead, it is applied to actual continuous outputs. Indeed, there is a weight for each classifier, and sometimes for each class and for each classifier. In the former case, there are \( T \) weights \( w_{1}, \ldots, w_{T} \) commonly acquired as estimates accurse from training, the total support of class \( w_j \) is defined in Equation (6).

\[
\mu_j(x) = \sum_{t=1}^{T} w_{tj} d_{tj}(x) 
\]

In the latter case, there are \( T \times C \) weights for class and classifier, thus the total support for class \( w_j \) is defined in Equation (7).

\[
\mu_j(x) = \sum_{t=1}^{T} w_{tj} d_{tj}(x) 
\]

where \( w_{tj} \) is the weight for \( t \)th classifier via classifying class \( oj \) patterns [17,48].

In some cases, there is a procedure applied on mean combiner when some of the classifiers give an unwanted output which may be high or low support to a particular class. This outlier result could be
avoided by removing the ensemble member which case this unwanted output before taking the mean. This procedure is called trimmed mean [41,48].

iii. Minimum/Maximum/Median Rule

As the name implies, these combination rules take the minimum, maximum, and median of the classifiers’ outputs as in Equations (8), (9), (10).

\[
\mu_j(x) = \max_{t=1,\ldots,T} d_{tj}(x) \quad (8)
\]

\[
\mu_j(x) = \min_{t=1,\ldots,T} d_{tj}(x) \quad (9)
\]

\[
\mu_j(x) = \text{median}(d_{tj}(x)) \quad (10)
\]

In all of the above equations, the final output ensemble is assigned to the class that holds larger support [54].

iv. Product Rule

In this combination rule, the support of each individual classifier is multiplied. Product rule is considered as very sensitive to the classifiers that characterized as pessimistic. Suppose that the support for a class is close to (0) this, this output will influence the overall ensemble system performance. But if all classifier’s outputs have produced estimated output correctly, then product rule offers the superlative estimation of the inclusive posterior probability of the final class selected via ensemble. The product rule id defined in Equation (11) [17,53].

\[
\mu_j(x) = \frac{1}{T} \prod_{t=1}^{T} d_{tj}(x) \quad (11)
\]

2) Decision Templates

The idea of the decision template (DT) is introduced by Kuncheva [53] which inspired form the idea of DP. The core idea is defining decision templates as the average DP observed for every class through training. For any given test example \(x\), its DP will be compared to the DTs of each class and the class whose DT is closest, according to selected similarity measure, is chosen as the final output class label. To clarify more accurately, the decision template for class \(\omega_j\), illustrated in figure 11, is calculated as in Equation (12) [17].

\[
DT_j = \frac{1}{N_j} \sum_{T_{j}} P_{X_{j} | \omega_j} DP(X_j) \quad (12)
\]

![Figure 11: DT related for class \(\omega_j\) obtained as averaged class \(\omega_j\) DPs [17]](image-url)
3) Dempster-Shafer Based Combination

Dempster-Shafer (DS) is a mathematical theory of evidence. It is a reasoning strategy with the uncertainty that can understand the relation to other frameworks like probability, possibility, and imprecise probability theories. The attractive feature of Dempster-Shafer theory is that it permits combining evidence from a different source (i.e. model’s outputs in the case of ensemble system) and reaches a degree of belief which takes into account all evidences that came from multiple sources [55,56] originally been employed in military application, like in sensor fusion of tracking target, or foe identification. It has the ability to combine any decision making problem like classification tasks. The key idea of using DS in combining ensemble is to interpret their outputs as a measure of evidence provided by constitute classifier s[57,58]

According to the DTs method, the c DTs, DT1, . . . , DTc are originate from the data. An alternative for calculating the similarity among the decision template DTi and the decision profile DP(x), the subsequent steps are carried out [17,48]:

i. Let $DT_j^i$ be the I th row of DTj, and Di(x) the output of the i th classifiers. that is, $Di(x) = [d_{i,1}(x), ..., d_{i,c}(x)]$ the i th row of the decision profile DP(x). We calculate the “proximity” $\Phi$ between $DT_j^i$ and the output of the classifier Di for the input x as in Equation (13).

$$\Phi_j(x) = \frac{(1+\|DT_j^i-D_i(x)\|^2)^{-1}}{\sum_{k=1}^{c}(1+\|DT_j^i-D_k(x)\|^2)^{-1}} \tag{13}$$

where $\|\|$ is any matrix norm. For example, we can use the Euclidean distance between the two vectors. Thus for each decision template we have L proximities.

ii. Using (11), we calculate for every class, j=1, . . . , c; and for every classifier, i = 1, . . . , L, the following belief degrees expressed in Equation (14).

$$b_j(D_i(x)) = \frac{\Phi_j(x)[1-\Phi_k(x)]}{1-(\Phi_j(x)[1-\Phi_k(x)])} \tag{14}$$

iii. The final degrees of support are expressed in Equation (15).

$$\mu_j(x) = K \prod_{i=1}^{L} b_j(D_i(x)), \quad j = 1, \ldots, c \tag{15}$$

where K is a normalizing constant.

4. IDSs Based on Clustering Ensemble

Clustering methods are considered as unsupervised learning approaches that bind a set of instances to groups according to the similar traits of these instances [59].

traits could be any type of measures like distance, probability, density [60]; for example,
Figure 12: Taxonomies of (a) clusters and (b) outliers [14].

consider that we have set of unlabeled instances, and suppose that these instances lie in two dimensions (X and Y), they could be grouped, for example, into five different clusters as shown in figure 12 (a). An outlier is another taxonomy of clustering methods in which some instances in the dataset characterized by high deviation than other instances which denoted as regular ones; for example, in figure 12 (b), instances O1 to O4 are considered as outliers whereas instances in N1 to N2 clusters are normal ones.

There are exist large spectrum of clustering techniques that could be utilized to develop an intrusion detection system, especially anomaly IDS. These techniques could be grouped into two distinct types: regular and co-clustering [61]. The key distinguished between these two clustering types is the strategies about how they dealt with the observations as well as features of the instances [62,63]. To clarify more accurately, regular clustering like K-means [64,65], K-medoids [66–68] collect instances from the observations of dataset while co-clustering instantly cooperate both observation and features of the dataset to produce clusters.

Clustering-based techniques have been used in various applications, and they find its sound in intrusion detection especially in anomaly detection. When clustering techniques utilized to develop anomaly detection systems, they are facing three different scenarios: The first one assume that normal instances often grouped in one cluster while attack instances are not. This means that in any anomaly detection system as clustering technique recognizes any instances that not belong to normal cluster, as attacks. This scenario hinders clustering techniques to detect anomalies in an optimized manner if there is a noise data instances clustered as anomalies as in [69]. The second scenario assumes that normal instances are always located near to the centroid of the cluster, in the opposite, attack instances, are often located far away from the centroid. A weak point of this scenario is that if a system considers that all far instances from the centroid as anomalies, it may fail to identify anomalies located near to the cluster centroid (which could be noise or outlier instances) [61]. The third scenario has been overcoming the limitation of the second one. In the third scenario, the assumption is that normal data instances are located in massive and intensive clusters, while attack instances located in small ones. Systems take in their consideration this scenario when developing an anomaly IDS have been identified instances and bind them to clusters according to size and/or densities [70].

However, the major drawback of clustering techniques is that; most of them can deal with just continuous attributes [14], there are exist difficulties to determine the optimal number of clusters [71],
besides, some clustering techniques face local optima trap [72]. Nevertheless, works in literature overwhelmed with attractive researches that have been utilized clustering techniques in an efficient way that oriented to various scopes.

There exists a massive number of works that employ clustering techniques for developing and implementing IDSs [73–78], however, to authors’ best knowledge, very few publications are available in the literature that addresses the issue of clustering ensemble for intrusion detection. The next paragraphs will highlight those attracting works that have been utilized ensemble clustering methods when developing their IDSs.

Yuan, Li, Yao and Zhang [79] proposed a new approach for detecting network anomalies based on the ensemble of clustering. Their approach relayed on deep learning and enable subspace spectral ensemble clustering to detect anomalies and they called it “DEP-SSEC”. Their proposed system overcomes the limitation of manually feature extraction that not take into consideration the good presentation of semantic features in URLs by cooperating word2vec method. DEP-SSEC operates in three steps. First, it utilizes an ensemble clustering model to discriminate normal and malicious instances, then it employs word2vec method for getting semantical of anomalies. Finally, it makes uses another multi-clustering method to identified specific attack types. Their proposed approach was evaluated on a real-life dataset called Qhoo 360.

Chen, Kong, Mei, Yuan and Li [80] provided a new clustering ensemble method to detect novel attacks. The proposed method is divided into three major components; the first one is feature selection, which transforms the dataset into subspaces. The second one includes four clustering ensemble models named DBSCAN, one-SVM, EM, and Agglomerative Clustering. Each of these models is used different subspaces to get clustering labels. Then the conducting labels will be fed into the voting model, which is the last component of the proposed clustering ensemble, to get the final result. NSL-KDD 2009 dataset is used to evaluate the proposed clustering ensemble.

Work done by [81] presented a new approach of ensemble clustering that operates in parallel to enhance the system’s speed and performance. The proposed system relies on Parallel Evidence Accumulation as well as ensemble clustering to detect network attacks and hence they called the system “PEA-IDS”: the base clustering ensemble model is K-means that operate in a number of distinct processors several times (according to predefine K). eventually, the output clusters of the ensemble will have contained normal and anomalies instances according to the dense of resultant clusters. KDDCup99 data set has been used for evaluation “PEA-IDS”.

Wankhade, Patka and Thool [82] argue that the clustering ensemble could increase the detection rate and at the same time decrease the false positive rate. Their proposed method rely on five main components to detect intrusion named feature selection, filtering, clustering, divide and merge, clustering ensemble. The core idea of their system is to use the k-means clustering algorithm several times, hence, overcome the limitation of chosen unappropriated k. After obtaining the clusters in the first round, they are divided and merge again and calculate several k cluster centroids. Eventually, each point in the filtered dataset will be assigned to specific centroid based on its density.

The results obtained by [83] suggest that the clustering ensemble with boosting method are effective for intrusion detection. The proposed algorithm has been adapted k-means model as an ensemble base learners and boosting as ensemble method. The suggested algorithm overcomes the two major limitations of k-means which are sensitive to noise, outlier as well as difficulties when handling huge data. In addition, the divide and merge method has been utilized for improving performance represented by detection rate and false-positive rate. Authors have employed windows method to handle dataset as junks and save the statistics of the data to be used whenever needed again during boosting. KDDCup’99 has been used for evaluating the proposed algorithm.

The most interesting work to this issue has been proposed by [84] when they utilized outlier, which is a branch of unsupervised learning, and ensemble concept to detect intrusion. Two outlier algorithms are utilized as ensemble constituted components; the former is based on a similar coefficient sum while the latter based on Kernel Density. The combination is done with the aid of the voting method. Authors have been used PCA for dimensionality reduction and utilized code mapping as preprocessing.
for character features. They chose KDD99 data set to evaluate their works in a term of detection rate and false-positive rate.

Weng, Jiang, Shi and Wu [85] developed a novel anomaly detection method using Evidence Accumulation (EA) based on the clustering ensemble hence called EADS. The core idea of their proposed system is to run k-means several times to form a hierarchical clustering tree and obtain a final partition of the dataset. In EADS, after running EA, a number of clusters have been generated. The assumption is that instances fall in the same cluster are with the same type. Eventually, after calculating the proportion of each cluster, the large cluster denoting as normal one, while others as anomalies. The effectivity of the proposed system is tested on KDD99 CUP dataset and compared with other clustering algorithms.

Ye, Li, Chen and Jiang [86], Xin Hu, and Kang G. Shin [87] have demonstrated that ensemble clustering could be employed effectively to detect malware families. In [86], the authors proposed a system called Automatic Malware Categ...
Table 2: Summary table of clustering-based ensemble for IDSs

| Ref | Method Name | Technique(s) | Combination method | Preprocessing | Dataset | Metrics |
|-----|-------------|--------------|--------------------|---------------|---------|---------|
| [79] | DEP-SSEC | subspace spectral ensemble clustering+ word2vec | Voting | Gaussian mixture model (GMM) and One Class Support Vector Machine (one-SVM) | Qihoo 360 | TPR, FPR |
| [80] | NEC | DBSCAN, one-SVM, EM, and Agglomerative Clustering | Voting | Normalization | NSL-KDD 2009 | DR, FPR |
| [81] | PEA-IDS | K-means | Evidence Accumulation (EA) | Normalization | KDDCup99 | DR, FPR, Speedup ratio (SR), and Parallel efficiency (PE). |
| [82] | - | K-means | Divide and merge | Features selection, filtering | - | - |
| [83] | - | K-means | Voting | windows method | KDDCup'99 | DR, FAR, Precision, recall, F-measure and G-mean |
| [84] | - | similar coefficient sum , and Kernel Density | Voting | PCA for dimensionality reduction and code mapping as preprocessing for character features | KDDCup'99 | DR, and FPR |
| [85] | EAIDS | K-means | Evidence Accumulation (EA) | Features selection | KDD99 CUP | DR, and FPR |
| [86] | AMCS | hybrid hierarchical clustering (HHC) algorithm with the aid of (TF-IDF) and (TF)+ weighted subspace K-medoids | Features selection | Kingsoft Anti-Virus Lab malware | Macro-F1 and Micro-F1 measures, |
| [88] | - | (LOF, COF, INFLO) for density-based+ (RBDA, RADA, and ODMR) for rank-based | Average, and min rules | Selection | KDD,99, Wisconsin, and PEC datasets | Rank Power(RP), AUC, and FP |
| [89] | - | KM-GSA, KM-PSO, FKM | WMV | Non | KDDCup99 | Accuracy, FP, TP |
| [90] | - | FKCM | ACO | Sampling and features selection | KDDCup99 | DR, ER |

5. IDSs Based on Classification Ensemble
In the last decade, the researchers have been directed their attention to machine learning and data mining for solving a large spectrum of problems in different aspects, and IDSs are one that considered a challenging problem in which the researchers have paid effort to develop and implement such systems. Firstly, these systems are initially concentrated on a single classifier in their core for accomplishing the desired goal such as a neural network (NN), support vector machine (SVM), and etc. However, the later studies have been shifted to employee an ensemble of classifiers for mitigating the imperfections of a single classifier model [29,91–93]. The success of the concept of ensemble classifiers have been assured both theoretically by Dietterich [29] as well as empirically by Miranda [94]. Nevertheless, several papers have indicated that the ensemble of classifiers could improve the overall accuracy compared with the single classifier model [29,93–96] In general, implementing an ensemble of classifiers will reduce the overall error rate.

To the best of the authors' knowledge, several publications have appeared in recent years documenting a survey in the field of IDSs based on ensemble learning. However, [26] , and [97] among these engaging surveys focusing on classification ensemble extensively. Both surveys posed common questions when designing an ensemble system that says:

- “How to generate a set of diverse individual classifiers?” [26]
- “How should suitable base components for an ensemble be created?” [97]
The above two questions point out a very important feature of ensemble systems which is called “Diversity”, and in this survey, we try to answer these questions carefully.

Diversity is considered as a “Cornerstone of Ensemble Systems” as Robi Polikar [17] said. If one gets a classification model with perfect generalization and performance, there is no need to introduce ensemble techniques. However, there are not exists perfect classifiers due to the fact of noise, outliers, and overlapping in the distribution of data. Researchers seek to find out classifiers that could classify the field of data most of the time correctly. This implies that each model makes an error in different patterns. The goal of this implication is that if each classifier makes a different error, then the fusion of these classifiers could reduce the total error. More precisely, the success of an ensemble system needs classifiers whose decision boundaries are sufficiently different from each other. An ensemble of such characteristic component is said to be diverse. According to [27,48] , one can enforce ensemble diversity via manipulation of either classifiers input, outputs, or models.

Previously we mentioned that the ensemble system could be either homogenous or heterogeneous based on its structure. The next subsection will focus on ensemble systems that adapted homogenous classifiers, while the later subsection will concentrate on ensemble systems that relying on heterogeneous classifiers.

5.1. IDSs Based on Homogenous Ensemble

The homogenous ensemble for IDSs has been launched in many works since it simple and can be employed straightforwardly and efficiently. For creating a homogenous ensemble, the learning algorithm is selected at first, then several variations of the selected algorithm are created and trained on a different subset of the original training dataset. Then the results of all ensemble constituted classifiers are aggregated and combined to obtain one reliable result. Here, in this subsection, we started by investigating several attractive works in the literature that focus on homogenous ensemble when developing and implementing IDSs in the period between (2009-2020).

Perdisci, Ariu, Fogla, Giacinto and Lee [98] presented an anomaly IDS called McPAD that detected an intrusion behavior in payload utilized multiple SVM classifiers to form the ensemble. Their proposed system counting on the statistical analysis method of payload named n-gram which is effective to detect anomalies in the payload. For evaluation McPAD, authors have been used two intrusion dataset named DARPA’99, and GATECH and employed clustering technique for dimensionality reduction. Voting, Averaging, Max, Min, Product are crucial combining rule that adapted in combing constituted classifiers of the proposed system.

Song, Takakura, Okabe and Kwon [99] suggested a new method to detect anomalies utilizing one-class SVMs as ensemble components and clustering technique (i.e K-means) for subspace feature selection. They showed that the proposed method has been succeeding in improving the detection rate and maintain a low false-positive rate. KDD Cup 1999 benchmark data set have been selected to evaluate their proposed method in a term of DR, and FPR that reflected via the ROC curve. Max's rule has adopted for combining the SVMs base learners.

Bagging, boosting, and stacking are three different ensemble methods that have been investigated by [100] when they suggested C4.5 as a base learner of an ensemble in their proposed IDS. NSL-KDD has been used for evaluation purpose in a term of TP, FP, Recall, and precision. For the stacking method, authors have been used the same model (i.e. C4.5) as a meta-learner that responsible for combining base learners.

Another solution for intrusion detection is described in [101] where they proposed an IDS based on ensemble concept. They selected hoeffding tree as a base learner with a boosting method for ensemble construction. KDDCup’99 dataset has been selected for evaluation in a term of accuracy. The authors obtained their results with all (41) dataset features.
The work conducted by [102] proposed Resilient Back Propagation (RBP) model for homogenous ensemble construction. In this work, the authors utilized different strategies named (WMV, Weighted Product Rule(WPR), and Neyman Pearson cost minimization strategy) for combining base ensemble models. Two benchmarks dataset named (KDD’99, and DARPA’99) and one synthetic (generated DDoS attack in author lap) dataset have used for evaluating the proposed system after conducting Normalization preprocessing. Accuracy, TP, FP, cost per sample, and ROC are evaluation metrics that have been adapted in the proposed ensemble system.

Another solution that used a C4.5 decision tree as a homogenous ensemble based learner to form IDS is described in [103]. Greedy Boosting technique has been utilized for ensemble construction and combining. The key idea of Greedy-boosting is that instead of taking the weighs of misclassified instances in last T iteration for involving them in the next training cycle, Greedy-boosting takes all previous iterations of misclassified instance in the next iteration T. Authors have been employed KDD’99 data set for evaluating the work and reach excellent result especially for DOS which reached 100% detection rate and reach excellent result especially for DOS which reached 100% detection rate.

Syarif, Zaluska, Prugel-Bennett and Wills [18] applied four different classification models named Naïve Bayes, iBK, Jrip, and J48 as a standalone model, then used the same models in bagging and boosting ensemble to determine with is better for intrusion detection in a tern of accuracy and false positive. The results obtained from bagging techniques were 89.57, 99.44%, 99.71, and 99.67% for Naïve Bayes, iBK, Jrip, and J48 respectively. Boosting has shown comparable results except for J48 that shows surprising result reached 99.80%.

Lin, Zuo, Yang and Zhang [104] derived an ensemble system based on SVMs models. They employed two-level ensemble combination strategies. The first one has utilized the win-take-all method, while the second is used majority vote. The random forest has been used as a sampling method for the KDDCup’99 dataset in the second level. For evaluation purposes, DR, FPR, and TPR have been utilized as evaluation metrics. In this work, the authors proposed that each base learner (i.e. SVM) is responsible for the detection of distinct types of intrusion (i.e. Normal Probe, U2R, R2L, D O S). they conclude that the second level of ensemble based on RF and MV is achieved better accuracy for R2L and Probe attacks.

Balon-Perin and Gambäck [105], [106], and [107] have been used different decision trees for the homogenous ensemble to detect intrusions in network traffic. The Majority vote and fuzzy combination based on tree cost and detection rate are utilized as an ensemble combination for the first and second work respectively. In [105], SVM, LGP, and ARS have been utilized as features selection. While in [106], features are selected via gain ratio. The third work has utilized a partial decision tree as an ensemble base learner and GA for feature selection form the NSL-KDD dataset. All work has been adopted accuracy, TP, FP as evaluation metrics. In addition, model building time has been employed in [107] as an extra evaluation metric.

Amini, Rezaeaeonour and Hadavandi [108] proposed an ensemble for intrusion detection based on RBF as a base learner. what is attractive in this work is the combination strategy that has been adapted which depends on a hybrid of WMV and winner-takes-all to make the final judgment on NSL-KDD dataset records. Fuzzy clustering is used for selecting the training subset. Accuracy, Precession, Recall, F-value are crucial metrics for evaluation purposes that have been adopted in this works.

Pham, Foo, Suriadi, Jeffrey and Lahza [109], have been proposed homogenous ensemble based on J48 decision trees as ensemble base learners to form IDS. Both of them have been used the NSL-KDD dataset for training, testing, utilized accuracy, FAR, ROC, and other interesting metrics for evaluation purposes. In [109], the leave-one technique has been used with the aid of the NB classifier for feature selection. While in [110] Correlation-based Feature Selection(Cfs) has utilized as features selection. Voting is a combination rule that has been adapted in combining the constituted classifiers. Table 3 have been summarized works in the period between (2009-2020) that focused on heterogeneous ensemble when designing and implementing IDSs taking into account several entries (Ref, Ensemble components, Combination method, Preprocessing, Dataset, and metrics for evaluation).
5.2. **IDSs Based on Heterogeneous Ensemble**

The heterogeneous ensemble is another way for constructing an ensemble constituted classifiers. In this ensemble architecture, a set of diverse and different kinds of classifiers is trained on the same training set. Each of them has its capabilities for classification tasks on the dataset, hence, they produce diverse and different outputs that eventually aggregated and combined with the aid of combination rule to obtain the final ensemble result. The major concern of such an ensemble method is that each expert (i.e. classification algorithm) utilized a particular technique for constructing the classification hypothesis, which in turn, produces different outputs. Hence, for obtaining the final ensemble result, the output of each hypothesis should be in an interpretable form. There are many strategies for combining the ensemble components and voting is considered as the simplest form of combination rules [26,97].

In this section, an IDSs based on heterogeneous ensemble is presented with particular focusing onto works in the period between (2009-2020). The major concentrating will be on the following points:

1) Ensemble components represented by what are the types of base classifiers that have been used in the work.
2) Clarify the combination strategies that have been adapted in combining the heterogeneous classifiers.
3) Illustrate the preprocessing techniques for the dataset if it is involved.
4) Specify what are the datasets that have been used for evaluation purposes
5) And finally, demonstrates the metrics used in the evaluation of heterogonous ensemble of IDSs

Chou, Fan, Fan and Makki [111] presented an IDS based on the heterogeneous ensemble. Three different base classifiers have been used for constructing the ensemble named NB, fuzzy KNN, and BPNN. KDD99 dataset has been used for training and testing the constructed ensemble after removed the redundant records from the dataset and selecting three different feature subsets. The authors have selected DR, FPR, and CR as evaluation metrics. The work revealed that the classification rate when used majority vote, Dempster-Shafer, average rule, and Bayesian rule as combination rules are 88.72%, 87.59%, 86.41%, and 92.75% respectively.

Another work presented by Zaina et al [112] is focused on creating a heterogeneous ensemble system based on LGP, ANFIS, and RF classifiers that combined using majority vote rule. The KDDCUP’99 dataset has been selected for evaluation purposes in a term of accuracy, FP, and TP and applied features selection using Rough-BPSO method. The empirical results have been shown improvements in detection accuracy for all classes that reached 99.27%, 99.88%, 98.26%, 99.96%, and 99.79% for Normal, Probe, DoS, U2R, and R2L respectively.

Liu, Chen and Hu [113] proposed IDS based on three diverse classifiers named BPNN, probabilistic NN, and SVM that combined using Dempster-Shafer evidence theory for detecting malicious programs. They employed 2358 normal program and 651 virus program to train the heterogeneous ensemble and reflect the evaluation results that reached 97.044%, and 0.9699 for accuracy and ROC metrics.

The results obtained by Govindarajan and Chandrasekaran [114] suggested that anomaly IDS could be generated with high generalization performance with the aid of heterogeneous classifiers named RBF, and MLP that combined via the weighted vote. Their method relied on KDD 1999 dataset for training and testing the generated ensemble system, and employed the accuracy metric for evaluation. In this work, the authors have been used GA and EMD for feature selection.

Boro, Nongpoh and Bhattacharyya [115] and Govindarajan and Chandrasekaran [116] have proposed IDS based heterogeneous classifiers. In [115], the authors have been utilized three diverse classifiers named C4.5, NB, and Decision template, and suggested a combination rule which is WMV for combining the output of algebra combination rules. They have been used information gain as a feature selection technique to extract interest feature from two datasets named KDD’99, and TUIDS.
port scan. The results obtained in a term of accuracy are 98.42% and 99.84% for KDD99 and TUIDS port scan respectively. While in [116], the authors have been utilized RBF and SVM as ensemble constituted classifiers that combined via voting. Preprocessing methods represented by normalization and BFS have been applied to the NSL-KDD dataset for feature selection. The accuracy metric is used for evaluation that recorded 85.19 % for the proposed ensemble method.

Meng and Kwok [117] proposed an architecture of false alarm filter utilizing heterogeneous ensemble based on voting for improving the accuracy and reducing the false alarm. Their architecture relied on four major components named data standardization, data storage, voted ensemble selection, and alarm filtration. SVM, J48, and KNN are selected as ensemble components. The work is evaluated on alarm data generated form Snort.

Chaurasia and Jain [118] and Govindarajan [119] developed and IDS based on an ensemble of different classifiers. In [118], the ensemble is constructed with the aid of NN and KNN classifiers that combined with MV combination rule. The KDD’99 datasets is used in the work for evaluation purposes and gets results represented by 0.8852, 0.021, 0.8888, and 0.9152 for TP, FP, Precession, and recall respectively. In the same context [119] employed RBF and SVM for constructing a heterogeneous ensemble that combined also with MV rule. Acer07, and NSL-KDD datasets utilized for training and testing the proposed system in a term of accuracy. The recorded results of the work are 99.90% and 98.46% for NSL-KDD and Acer07 datasets respectively.

Tama and Rhee [120] have been presented IDS based on an ensemble of three classifiers named C4.5, Random Forest, and CART. Two combination rules have been applied to the proposed ensemble which is MV, and average of probabilities rules. The practical swarm optimization method has been applied to the NSL-KDD dataset for selecting the best attributes. The authors have been adapted three different number practical that are 50,100, and 200. The experimental results have been shown that the ensemble of the selected classifiers combined via MV and average of probabilities with PSO=50 are 99.76 and 99.80 while if PSO=100 the results are 99.64 and 99.76. The worst results have been obtained when PSO=200 that recorded 99.44 and 99.39 when combined with via MV and average of probabilities respectively.

Aburomman and Ibne Reaz [39] proposed an ensemble system based on two heterogeneous classifiers called SVM and KNN that combined using WNV rule. The proposed system has been applied to KDD’99 datasets. In this work, six different SVM obtained by variation of RBF values and six different values of K parameters in KNN. The dataset is subdivided into 5 datasets form training and testing the proposed system. The average result of the ensemble with WMV is 65.794%, 96.2464%, 98.4448%, 99.7939%, and 82.897% for normal, probe, DoS, U2R, and R2L respectively.

Jabbar, Srinivas and Sai Satyanarayana Reddy [121] proposed the IDS system based on a combination of ADTree and NB classifiers. The final result of the ensemble is obtained via MV combination rule. Discretization is a preprocessing method that applied to the NSL-KDD dataset. The experimental results have shown that detection accuracies are 100%, 99.92% for DOS and Probe respectively, and 91.41% for U2R and R2L.

Again Jabbar, Aluvalu and Reddy [122] presented an attractive heterogeneous ensemble based on two models named Random Forest (RF) and Average One-Dependence Estimator (AODE) that combined via voting. The authors have been made Numeric to binary preprocessing to the Kyoto dataset. The evaluation result of the proposed system has been recorded accuracy of 90.51% and FAR of 0.14.

Rajasekaran and Ayyasamy [123] developed an heterogenous ensemble for IDS based on EMSVM, KNm and IPSO which are combined via voting combination rule. Three features section are adapted called Intelligent Agent based Attribute Selection Algorithm (IAASA), IGR, and Intelligent Conditional Random Field (CRF) based Feature Selection Algorithm ICRFSSA for KDD’99 features selection. The authors relied on accuracy metric to evaluate their developed system that reported better accuracy compared with single classification model.

Salo, Nassif and Essex [124] presented IDS based heterogeneous ensemble with three constituted classifiers named SVM, IBK, and MLP. Voting is used for combining the ensemble components and
obtaining the final result. In this work, IG and PCA are utilized for feature selection techniques. Three benchmark datasets are selected for evaluating the proposed ensemble system named ISCX 2012, NSL-KDD, and Kyoto 2006+. Several vital evaluation metrics are used for testing the performance of the purpose work which recorded 99.01, 0.991, 0.010, 0.991, 0.992, 2, 3.49 for accuracy DR, FAR, Precision, F-Measure, building time (s) testing time (s) respectively.

The work presented by Ludwig [125] utilized heterogeneous neural network models to form the ensemble for intrusion detection which are combined via the majority voting method. The author employed PCA as a visualization preprocessing method applied to the NSL-KDD dataset. The experimental results of binary classification (i.e. normal and attack) recorded 92.50%, 91.62%, 14.72%, 97.95%, 93.70% for accuracy, AUC, FAR, DR, and F1 score respectively.

Zhou, Cheng, Jiang and Dai [126] proposed an ensemble method based on combining C4.5, Random Forest (RF), and Forest by Penalizing Attributes (Forest PA) using the voting method and other algebraic combination rules. Besides, the authors proposed a heuristic features selection method called CFS-BA for dimensionality reduction and selecting the best subset features. The experimental results of the proposed work are done using NSL-KDD, AWID, and CIC-IDS2017 datasets. The best results obtained for the NSL-KDD with 10 feature selected via CFS-BA and voting method were 99.81, 99.8, and 0.08. While in AWID-CLS-R dataset with 8 feathers selected were 99.52, 99.5, and 0.15, finally for CIC-IDS2017(Wed.) with 13 feathers selected were 99.89, 99.9, and 0.12 for accuracy, DR, and FAR respectively.

The most interesting approach to this issue has been proposed by Tama et al [127] when they proposed anomaly-based IDS utilizing the stacking ensemble concept of heterogeneous models represented by RF, gradient boosting machine, and XGBoost. In this stacking ensemble, the authors have been selected the Generalized Linear Model (GLM) as a stacker (i.e. meta-learner) that responsible for combining the ensemble constituted classifiers and obtaining the final result. The proposed stacking ensemble has been reported accuracy results of 99.9882, 89.5217, 92.1664, and 93.6256 for CSIC-2010v2, CICIDS-2017, NSL-KDD, and UNSW NB-15 datasets respectively. Table 4 have been summarized works in the period between (2009-2020) that focused on heterogeneous ensemble when designing and implementing IDSs taking into account several entries (Ref, Ensemble components, Combination Rule, Preprocessing, Dataset(s), and evaluation metric(s)).

Table 3: Summary table of homogeneous ensemble for IDSs

| Ref     | Ensemble components | Combination Rule         | Preprocessing          | Dataset(s)   | Evaluation metric(s) |
|---------|---------------------|--------------------------|------------------------|--------------|----------------------|
| [98]    | SVMs                | Voting, Averaging, Max, Min, Product | Clustering features | DARPA'99, GATECH | ROC, AUC, FP         |
| [99]    | SVMs                | Maximum rule             | Clustering, feature selection | KDDCup'99    | ROC Curve            |
| [100]   | C4.5                | Bagging (voting) + Boosting(Voting) + Stacking (C4.5) | None | NSL-KDD | TP, FP, Recall, Precision |
| [101]   | hoeffding tree      | Boosting                 | Features selection | KDDCup'99 | Accuracy              |
| Authors | Methodology | Ensemble Type | Data Sets | Evaluation Metrics |
|---------|-------------|---------------|-----------|--------------------|
| [102]   | Resilient Back Propagation (RBP) | WMV, Weighted Product Rule (WPR), and Neyman Pearson cost minimization strategy | Normalization | KDD’99, DARPA’99, and generated DDoS attack in author lap | Accuracy, TP, FP, Cost per sample, ROC |
| [103]   | C4.5 Boosting | None | KDD Cup’99 | Precession, Recall |
| [18]    | Naïve Bayes, iBK, Jrip, K48 | Bagging + boosting + Stacking \(^2\) | None | NSL-KDD | Accuracy, FP, and execution time |
| [104]   | SVM | 1st ensemble level use win take all 2nd ensemble level use majority voting | Sampling via Random Forest | KDD Cup’99 | DR, FPR, TPR |
| [128]   | SVCs | Selection | PCA, Kernel PCA, Isometric Mapping Isomap | NSL-KDD | Accuracy, TP, FP |
| [105]   | DT | Majority vote | SVM, linear genetic programming (LGP), and multivariate adaptive regression splines (MARS) for features selection | KDD’99 | Accuracy, FP, FN |
| [106]   | J48 tree | Fuzzy combination based on tree cost and detection rate | Gain ratio feature selection | KDD’99 | Accuracy, Cost |
| [107]   | Partial Decision Tree | Vote | Genetic algorithm | NSL-KDD | Accuracy, TP, FP, and model building time |
| [108]   | Radial Basis Function (RBF) | hybrid of WMV and winner-takes-all | Fuzzy clustering subset training set selection | NSL-KDD | Accuracy, Precession, Recall, F-value |
| [129]   | C4.5 Bagged tree, AdaBoost, LogitBoost, GentleBoost, RUSBoost(Vote) | None | UNSW-NB15 | Accuracy, ROC, AUC, TP, FP |
| [109]   | J48 | Bagging and Boosting (Vote) | “leave-one-out” techniques and Naïve Bayes classifier for feature selection | NSL-KDD | Accuracy, FAR |
| [110]   | J48 | AdaBoost, Bagging, and Stacking | Cfs, Chi-square, SU, Gain Ratio, Info Gain, and OneR used for feature selection | NSL-KDD | TP rate, FP rate, Precision, Recall, ROC, Accuracy |
| [130]   | CART | AdaBoost | GFR features reduction, average MCC as Subset features selection | KDD’99 | Accuracy |
| [131]   | SVMs | Stacking (SVM) | Normalization | NSL-KDD | Accuracy, Detection Rate (DR), False Alarm Rate (FAR) |
| [132]   | extremely randomized trees | Bagging | Transformation | KDD Cup’99 and NSL-KDD | Precision, Recall, F1-score, Support |
| [133]   | SVMs | Majority Voting | Feature Selection based on Extra Trees | Cambridge | Accuracy, F-score |

---

\(^2\) Authors applied Stacking with is considered as heterogeneous ensemble method.

**Table 4:** Summary table of heterogeneous ensemble for IDSs
| Author(s) | Ensemble components | Combination Rule | Preprocessing | Dataset(s) | Evaluation metric(s) |
|-----------|---------------------|-----------------|--------------|------------|---------------------|
| [111]     | NN, Fuzzy KNN, Naive Bayes classifier | majority voting, average rule, Dempster-Shafer technique, and Bayesian combination method | Remove redundant records | KDD99 | Detection rate (DR), false positive rate (FPR), and classification rate (CR). |
| [112]     | LGP, ANFIS, and RF | Weighted Voting | Features selection using Rough-BPSO | KDD Cap 1999 | Accuracy, FP, TP |
| [113]     | BP neural network, probabilistic neural network and SVM | Dempster-Shafer evidence theory | None | 2358 normal programs and 651 virus programs | ROC |
| [114]     | MLP, RBF | Voting | GA and EMD based feature selection | KDD 1999 | Accuracy, precision, recall, training time |
| [115]     | NN, C4.5 | Voting | Genetic algorithm | KDD’99 | TP, FP, Precision, Recall and F-measure |
| [116]     | C4.5, Naive Bayes, and Decision Table | weighted majority Voting | Information Gain | KDD’99, TUIDS Port Scan Dataset | Accuracy |
| [117]     | SVM, J48, and KNN | Voting | Normalization, BFS for feature selection | NSL-KDD | Accuracy |
| [118]     | NN, KNN | Majority Voting | None | KDD’99 | Accuracy, TP, FP, FPR |
| [119]     | RBF and SVM | Majority Voting | None | Acer07, NSL-KDD | Accuracy |
| [120]     | C4.5, Random Forest, and CART | Majority Voting, Average of Probabilities Rule | article swarm optimization (PSO) +CFS for attribute selection | NSL-KDD | Accuracy, FPR |
| [135]     | Bayesian Network, Naive Bayes and J48 | Majority Voting | Best First, Genetic Search and Rank Search based features selection | NSL-KDD | TPR, FPR, FMeasure, ROC Area, Time for build model |
| [39]      | SVMs, KNNs | WMV | Mapping, normalization | KDD 1999 | Elapsed time, Accuracy |
| [136]     | J48 and Random Tree, (RandomForest and Random Tree), and Random Forest Correct | Voting | None | KDD’99 | Attack detection rate and false attack detection rate |
| [121]     | ADTree, NB | Majority voting | Discretization | NSL-KDD | Accuracy, FAR, DR |
| [122]     | Random Forest(RF), Average One-Dependence Estimator (AODE) | Voting | Pre-processing Numeric to binary | Kyoto | Accuracy, DR, FAR, Hubert Index (HI) |
| [123]     | EMSVM, k-NN, and IPSO | Voting | Intelligent Agent based Attribute Selection Algorithm (IAASA), IGR, and ICRFFSA for features selection | KDD’99 | Accuracy |
| [124]     | SVM, IBK, MLP | Voting | RF, PCA | ISCX 2012, NSL-KDD, and Kyoto 2006+ | Accuracy DR, FAR, Precision, F-Measure, Building time (s) Testing time(s) |
| [125]     | AE, DBN, DNN, and ELM | Majority voting | PCA | NSL-KDD | DR, FAR, Precision, Recall F1-score, and Confusion matrix |
| [126]     | C4.5, RF, and Forest PA | Voting | CFS-BF for dimensionality reduction | NSL-KDD, AWID, and CIC-IDS2017 | Accuracy Precision DR F-Measure ADR FAR MBT(s) |
| [137]     | RF, LG, and KNN | Stacking (SVM metal earner) | RF and hashing for features selection | UNSW NB-15 packet-based, and UGR’16 flow-based datasets | Accuracy, Precision, Recall, F1 score, FAR, FPR, FN, R |
| [127]     | RF, gradient boosting machine, and XGBoost | Stacking (Generalized Linear Model (GLM)) | None | CSIC-2010v2, CICIDS-2017, NSL-KDD, and UNSW NB-15 | Accuracy, FAR, Confusion Matrix |
6. Conclusion

Despite many methods of knowledge extraction and classification, ensemble systems still one of the vital attractive and active areas in classification scope especially in IDSs. The extensive analysis of ensemble systems reveals that most of them have been successfully implemented and achieved excellent results in the intrusion detection field. This success is due to two main reasons, the first one relays on the fact some ensemble methods reduce the variance factor such as bagging or boosting, while the second one counts on minimizing bias like stacking ensemble method. Besides, there are also methods, such as cascading, which involve generating a new set of features based on the probability of estimation. In such cases, each ensemble constituted classifier has handled a portion of training data, while the rest delivered to other classifiers in the ensemble.

From our observations, there are many methods for combining and aggregating the outputs of ensemble classifiers to obtain the final result. These methods are very important since they aim to make a complementary between ensemble constituted classifiers. In a simple word, if one classifier makes an error for a specific output, this error could be corrected by other classifiers using an appropriate combination method. Although there are many combination strategies exits (we demonstrated them in Ensemble Combination Methods section), voting based combination methods are among a more used combination strategy. Weighted majority voting is another variation of voting that can be applied if the reliability of each ensemble experts could be estimated in advance via decelerate weight coefficients, hence, improving the overall performance of the ensemble system.

In this survey, we aimed to highlight two main types of ensemble system for IDSs

1) IDSs based clustering ensemble
2) IDSs based classification ensemble

For the first category, we have noted that there is a limited number of works in the literature that adopted the clustering ensemble for solving intrusion detection issues. This is because clustering-based methods are considered unsupervised methods which means that there are no classes are defined in advance. The process of devising classes for real traffic accurately is not easy. Hence, authors have been seeking their attention to classification ensemble for mitigating intrusion detection issues. Nevertheless, the clustering ensemble will still vital methods that are concerned with intense interest by intrusion detection system developers because of its features that include speed and reducing storage space during processing.

For the second category, we have been reviewed two major types of IDS based ensemble. The first one is the homogenous ensemble in which the base learners are induced using the same algorithm, whereas the second one is a heterogeneous ensemble in which base learner is induced by different learning algorithms. For homogenous ensemble, we have noted that almost the works in the literature are adapted three main learning methods named (SVMs, DTs, and NNs). Whereas in a heterogeneous ensemble, various types of learning algorithms are employed to form the ensemble, nevertheless, SVM, DT, and NN are the most technologies that have been employed.

Moreover, the authors have been noted that a large percentage of research studies are evaluated using the KDD’99 dataset. However, this dataset inherent limitations and weaknesses for presenting the perspective of network real traffic thus is considered as an outdated dataset. Some researchers have been seeking their works to NSL-KDD that solve few problems inherent in the KDD’99 dataset, but still, these two datasets versions considered outdated datasets. Nevertheless, a large number of studies in the scope of
ensemble-based IDSs have been used these datasets for evaluation purposes. Some surveys in the same context commented on the use of this dataset (i.e. KDD’99) is due to the fact that there is no up to date dataset are available. However, from our perspective view, we totally disagree with this assumption, since there are may up to data datasets available in the literature like (CICIDS2017, and ADFA-IDS). However, we argue that the main reason for using the KDD’99 dataset (which is the same reason for hindering using new emerging datasets) is due to the fact that a large spectrum of works in the literature are used this dataset which make it benchmark testbed for researchers to compare their works. We strongly recommend researchers in the field of IDS to adapt the most recent and up to datasets summarized in [138,139], but taking into account that these datasets are less explored in the literature for evaluating IDSs. However, it could be utilized to form an effective IDSs.

From this survey, it can also see that ensemble classifiers are rarely adopted based on stacking for intrusion detection tasks, particularly for the heterogeneous ensemble. Therefore, researchers may explore the stacking ensemble and try to find out new methods for implementing their stacker (i.e. meta-learner) to implement an effective IDSs. Future works can be focused on developing a new method of stacker that could be a hybrid stacker paradigm. It is also observed that some IDSs based on ensemble employee optimization methods and this adds some trade-off to the user for selecting the best and appropriate solution especially for selecting the best subset of attributes or best coefficients weights for WMV.

7. References

[1] Homoliak I 2017 Intrusion Detection in Network Traffic (Dissertation, Faculty of Information Technology, BRNO UNIVERSITY OF TECHNOLOGY)
[2] Lin W C, Ke S W and Tsai C F 2015 CANN: An intrusion detection system based on combining cluster centers and nearest neighbors Knowledge-Based Syst. 78 13–21
[3] Anon 2019 Internet Security Threat Report(ISTR) 61
[4] Mehd EBady 2015 Dynamic DDoS Attack Detection based on Data Mining Approach (College of Information Technology Software Department,University of Babylon)
[5] Elbashony R M, Sallam E A, Eltobely T E and Fahmy M M 2013 A hybrid network intrusion detection framework based on random forests and weighted k-means Ain Shams Eng. J. 4 753–62
[6] Ni X, He D, Chan S and Ahmad F 2016 Network anomaly detection using unsupervised feature selection and density peak clustering Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) vol 9696 pp 212–27
[7] Wang Y, Meng W, Li W, Liu Z, Liu Y and Xue H 2019 Adaptive machine learning-based alarm reduction via edge computing for distributed intrusion detection systems Concurr. Comput. 31 1–12
[8] Folino G, Pizzuti C and Spezzano G 2010 An ensemble-based evolutionary framework for coping with distributed intrusion detection Genet. Program. Evolvable Mach. 11 131–46
[9] Ni X, He D and Ahmad F 2016 Practical Network Anomaly Detection Using Data Mining Techniques VFAST Trans. Softw. Eng. 9 1
[10] Mohammad M N, Sulaiman N and Muhsin O A 2011 A novel Intrusion Detection System by using intelligent data mining in WEKA environment Procedia Comput. Sci. 3 1237–42
[11] Roesch M 2015 Snort – Lightweight Intrusion Detection for Networks Lisa vol 99 pp 229–38
[12] Brahma A and Panigrahi S 2020 Database Intrusion Detection Using Adaptive Resonance Network Theory Model Advances in Intelligent Systems and Computing vol 990 (Springer) pp 243–50
[13] Lim S Y and Jones A 2008 Network anomaly detection system: The state of art of network behaviour analysis Proceedings - 2008 International Conference on Convergence and Hybrid Information Technology, ICHIT 2008 pp 459–65
Bhuyan M H, Bhattacharyya D K and Kalita J K 2017 Network Traffic Anomaly Detection Techniques and Systems (Springer)

Buczak A L and Guven E 2016 A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection IEEE Commun. Surv. Tutorials 18 1153–76

Kwon D, Kim H, Kim J, Suh S C, Kim I and Kim K J 2017 A survey of deep learning-based network anomaly detection Cluster Comput. 22 1–13

Polikar R 2006 Ensemble based systems in decision making IEEE Circuits Syst. Mag. 6 21–44

Syarif I, Zaluska E, Prugel-Bennett A and Wills G 2012 Application of bagging, boosting and stacking to intrusion detection vol 7376 LNAI

Zhou Z H 2012 Ensemble methods: Foundations and algorithms (Chapman and Hall/CRC)

Dasarathy B V. and Sheela B V. 1979 A composite classifier system design: Concepts and methodology Proc. IEEE 67 708–13

Hansen, L.K. and Salamon P 1990 Neural network ensembles. IEEE transactions on pattern analysis and machine intelligence 12(10) pp.993-1001

Schapire R E 1990 The Strength of Weak Learnability Mach. Learn. 5 197–227

Jacobs R A, Jordan M I, Nowlan G E 1991 Adaptive mixtures of local experts Neural Comput. 3 79–87

Wolpert D H 1992 Stacked generalization Neural networks 5 241–59

Xu L, Krzyszak A and Suen C Y 1992 Methods of combining multiple classifiers and their applications to handwriting recognition IEEE Trans. Syst. Man. Cybern. 22 418–35

Aburomman A A and Reaz M B I 2017 A survey of intrusion detection systems based on ensemble and hybrid classifiers Comput. Secur. 65 135–52

Woźniak Michal and Graña M and Corchado E 2014 A survey of multiple classifier systems as hybrid systems Inf. Fusion 16 3–17

Sesmero M P, Ledezma A I and Sanchis A 2015 Generating ensembles of heterogeneous classifiers using Stacked Generalization Wiley Interdiscip. Rev. Data Min. Knowl. Discov. 5 21–34

Dietterich T G 2000 Ensemble methods in machine learning International workshop on multiple classifier systems pp 1–15

Bagui S C 2005 Combining Pattern Classifiers: Methods and Algorithms vol 47

Rokach L 2010 Ensemble-based classifiers Artif. Intell. Rev. 33 1–39

TAN P-N, Steinbach M and Kumar V 2006 Pang-Ning Tan - Introduction to data mining

Ho T K 1998 The random subspace method for constructing decision forests IEEE Trans. Pattern Anal. Mach. Intell. 20 832–44

Dietterich T G and Bakiri G 1994 Solving multiclass learning problems via error-correcting output codes J. Artif. Intell. Res. 2 263–86

Chali Y, Hasan S A and Mojahid M 2014 Complex question answering: homogeneous or heterogeneous, which ensemble is better? International Conference on Applications of Natural Language to Data Bases/Information Systems pp 160–3

Kittler J, Hatef M, Duin R P W and Matas J 1998 On combining classifiers IEEE Trans. Pattern Anal. Mach. Intell. 20 226–39

Sindh S S S, Geetha S and Kannan A 2012 Decision tree based light weight intrusion detection using a wrapper approach Expert Syst. Appl. 39 129–41

Webb A R and Copsey K D 2011 Statistical Pattern Recogniton (WILEY)

Aburomman A A and Ibne Reaz M Bin 2016 A novel SVM-kNN-PSO ensemble method for intrusion detection system Appl. Soft Comput. J. 38 360–72

Webster J G, Nanni L, Brahnam S and Lumini A 2015 Classifier Ensemble Methods Wiley Encyclopedia Electron. Eng. 1–12

Han J, Kamber M and Pei J 2012 Data Mining: Concepts and Techniques (Elsevier)

Bauer E and Kohavi R 1999 An empirical comparison of voting classification algorithms: Bagging, boosting, and variants Mach. Learn. 36 105–39

Kearns M 1988 Thoughts on hypothesis boosting Unpubl. Manuscr. 45 105

Hatami N 2012 Some proposals for combining ensemble classifiers (Università degli Studi di
Cagliari)

[45] Freund Y, Schapire R and Abe N 1999 A short introduction to boosting Journal-Japanese Soc. Artif. Intell. 14 1612
[46] Ioannis Hatzilygeroudis V P (eds. . 2018 Advances in Hybridization of Intelligent Methods: Models, Systems and Applications (Springer International Publishing)
[47] Graczyk M, Lasota T, Trawiński B and Trawiński K 2010 Comparison of bagging, boosting and stacking ensembles applied to real estate appraisal Asian conference on intelligent information and database systems pp 340–50
[48] Kuncheva L I 2014 Combining pattern classifiers: methods and algorithms (John Wiley & Sons)
[49] Re M and Valentini G 2012 Ensemble methods: A review Advances in Machine Learning and Data Mining for Astronomy pp 563–94
[50] Association I R M 2013 Bioinformatics: Concepts, Methodologies, Tools, and Applications vol 1 (IGI Global)
[51] Huang Y S and Suen C Y 1993 The behavior-knowledge space method for combination of multiple classifiers IEEE computer society conference on computer vision and pattern recognition p 347
[52] Huang Y S and Suen C Y 1995 A method of combining multiple experts for the recognition of unconstrained handwritten numerals IEEE Trans. Pattern Anal. Mach. Intell. 17 90–4
[53] Kuncheva L I, Bezdek J C and Duin R P W 2001 Decision templates for multiple classifier fusion: an experimental comparison Pattern Recognit. 34 299–314
[54] GUNES V, MENARD M, LOONIS P and PETIT-RENAUD S 2003 Combination, Cooperation and Selection of Classifiers: a State of the Art Int. J. Pattern Recognit. Artif. Intell. 17 1303–24
[55] Wang X, Wang Y and Sun H 2016 Exploring the Combination of Dempster-Shafer Theory and Neural Network for Predicting Trust and Distrust ed M Graña Comput. Intell. Neurosci. 2016 5403105
[56] Shafer G 1976 A mathematical theory of evidence vol 42 (Princeton university press)
[57] Rogova G 1994 Combining the results of several neural network classifiers Neural networks 7 777–81
[58] Lu Y 1996 Knowledge integration in a multiple classifier system Appl. Intell. 6 75–86
[59] Sangaiah A K, Fakhry A E, Abdel-Basset M and El-henawy I 2019 Arabic text clustering using improved clustering algorithms with dimensionality reduction Cluster Comput. 22 4535–49
[60] Qin X, Ting K M, Zhu Y and Lee V C S 2019 Nearest-neighbour-induced isolation similarity and its impact on density-based clustering Proceedings of the AAAI Conference on Artificial Intelligence vol 33 pp 4755–62
[61] Chandola V, Banerjee A and Kumar V 2009 Anomaly detection: A survey ACM Comput. Surv. 41 1–58
[62] Bhuyan M H, Bhattacharyya D K and Kalita J K 2013 Network anomaly detection: methods, systems and tools Ieee Commun. Surv. tutorials 16 303–36
[63] Ahmed M, Naser Mahmood A and Hu J 2016 A survey of network anomaly detection techniques J. Netw. Comput. Appl. 60 19–31
[64] Li X, Li Y, Ling S, Strohmer T and Wei K 2020 When do birds of a feather flock together? k-means, proximity, and conic programming Math. Program. 179 295–341
[65] Fränti P and Sieranoja S 2018 K-means properties on six clustering benchmark datasets Appl. Intell. 48 4743–59
[66] Wang T, Li Q, Bucci D J, Liang Y, Chen B and Varshney P K 2019 K-medoids clustering of data sequences with composite distributions IEEE Trans. Signal Process. 67 2093–106
[67] Modak S, Chattopadhyay T and Chattopadhyay A K 2020 Unsupervised classification of eclipsing binary light curves through k-medoids clustering J. Appl. Stat. 47 376–92
[68] Kerstens E 2020 Non-Exhaustive, Overlapping k-medoids for Document Clustering Proceedings of the 53rd Hawaii International Conference on System Sciences
[69] Li H 2010 Research and implementation of an anomaly detection model based on clustering
analysis 2010 International Symposium on Intelligence Information Processing and Trusted Computing pp 458–62
[70] Moustafa N, Hu J and Slay J 2019 A holistic review of Network Anomaly Detection Systems: A comprehensive survey J. Netw. Comput. Appl. 128 33–55
[71] Melgarejo M, Rodriguez C, Mayorga D and Obregón N 2020 Time Series from Clustering: An Approach to Forecast Crime Patterns Recent Trends in Artificial Neural Networks ed A Sadollah and C M Travieso-Gonzalez (Rijeka: IntechOpen)
[72] Mehrotra S, Kohli S and Sharan A 2019 An intelligent clustering approach for improving search result of a website Int. J. Adv. Intell. Paradig. 12 295–304
[73] Yang Y, Zheng K, Wu C, Niu X and Yang Y 2019 Building an effective intrusion detection system using the modified density peak clustering algorithm and deep belief networks Appl. Sci. 9 238
[74] Pandeeswari N and Kumar G 2016 Anomaly detection system in cloud environment using fuzzy clustering based ANN Mob. Networks Appl. 21 494–505
[75] Tahir H M, Sa'd A M, Osman N H, Zakaria N H, Sabri P N M and Katuk N 2016 Oving K-means clustering using discretization technique in network intrusion detection system 2016 3rd International Conference on Computer and Information Sciences (ICCOINS) pp 248–52
[76] Harish B S and Kumar S V A 2017 Anomaly based Intrusion Detection using Modified Fuzzy Clustering. IJIMAI 4 54–9
[77] Bhuushan B and Sahoo G 2019 A Hybrid Secure and Energy Efficient Cluster Based Intrusion Detection system for Wireless Sensing Environment 2019 2nd International Conference on Signal Processing and Communication (ICSPC) pp 325–9
[78] Zhou M, Han L, Lu H and Fu C 2020 Intrusion Detection System for IoT Heterogeneous Perceptual Network Mob. Networks Appl. 1–14
[79] Yuan G, Li B, Yao Y and Zhang S 2017 A deep learning enabled subspace spectral ensemble clustering approach for web anomaly detection Proceedings of the International Joint Conference on Neural Networks vol 2017-May pp 3896–903
[80] Chen W, Kong F, Mei F, Yuan G and Li B 2017 A Novel Unsupervised Anomaly Detection Approach for Intrusion Detection System 2017 IEEE 3rd international conference on big data security on cloud (bigdatasecurity), IEEE international conference on high performance and smart computing (hpsc), and IEEE international conference on intelligent data and security (ids) pp 69–73
[81] Gao H, Zhu D and Wang X 2010 A Parallel Clustering Ensemble Algorithm for Intrusion Detection System 2010 Ninth International Symposium on Distributed Computing and Applications to Business, Engineering and Science pp 450–3
[82] Wankhade K, Patka S and Thool R 2013 An efficient approach for Intrusion Detection using data mining methods 2013 International Conference on Advances in Computing, Communications and Informatics (ICACCI) pp 1615–8
[83] Wankhade K K and Jondhale K C 2019 An ensemble clustering method for intrusion detection Int. J. Intell. Informatics 7 112–40
[84] Huang B, Li W, Chen D and Shi L 2009 An Intrusion Detection Method Based on Outlier Ensemble Detection 2009 International Conference on Networks Security, Wireless Communications and Trusted Computing vol 2 pp 600–3
[85] Weng F, Jiang Q, Shi L and Wu N 2007 An Intrusion Detection System Based on the Clustering Ensemble 2007 International Workshop on Anti-Counterfeiting, Security and Identification (ASID) pp 121–4
[86] Ye Y, Li T, Chen Y and Jiang Q 2010 Automatic Malware Categorization Using Cluster Ensemble Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining KDD ’10 (New York, NY, USA: Association for Computing Machinery) pp 95–104
[87] Hu X and Shin K G 2013 DUET: Integration of Dynamic and Static Analyses for Malware Clustering with Cluster Ensembles Proceedings of the 29th Annual Computer Security Applications Conference ACSAC ’13 (New York, NY, USA: Association for Computing Machinery) pp 79–88
[88] Zhao Z, Mehrotra K G and Mohan C K 2015 Ensemble Algorithms for Unsupervised Anomaly Detection Current Approaches in Applied Artificial Intelligence ed M Ali, Y S Kwon, C-H Lee, J Kim and Y Kim (Cham: Springer International Publishing) pp 514–25

[89] Bengdara S, NGADI M D A, SHARIF J M, ALI S, Sangeetha S V T, Ravi S, RajaRam U, Rammohan S R, AWANG N H, SHAHIBUDDIN S and others 2014 Ensemble of clustering algorithms for anomaly intrusion detection system J. Theor. Appl. Inf. Technol. 70 425–31

[90] Chen L W 2015 Network intrusion detection model with clustering ensemble method Int. J. Secur. its Appl. 9 249–50

[91] Khreich W, Granger E, Miri A and Sabourin R 2012 Adaptive ROC-based ensembles of HMMs applied to anomaly detection Pattern Recognit. 45 208–30

[92] Kotsiantis S B, Zaharakis I and Pintelas P 2007 Supervised machine learning: A review of classification techniques Emerg. Artif. Intell. Appl. Comput. Eng. 160 3–24

[93] Tsoumakas G, Partalas I and Vlachavas I 2008 A taxonomy and short review of ensemble selection Workshop on Supervised and Unsupervised Ensemble Methods and Their Applications pp 1–6

[94] Miranda Dos Santos E 2008 Static and dynamic overproduction and selection of classifier ensembles with genetic algorithms (École de technologie supérieure)

[95] Axelsson S 2000 Intrusion detection systems: A survey and taxonomy

[96] Kumar G, Kumar K and Sachdeva M 2010 The use of artificial intelligence based techniques for intrusion detection: a review Artif. Intell. Rev. 34 369–87

[97] Kumar G, Thakur K and Ayyagari M R 2020 MLEsIDSs: machine learning-based ensembles for intrusion detection systems—a review J. Supercomput.

[98] Perdisci R, Ariu D, Fogla P, Giacinto G and Lee W 2009 McPAD: A multiple classifier system for accurate payload-based anomaly detection Comput. Networks 53 864–81

[99] Song J, Takakura H, Okabe Y and Kwon Y 2009 Unsupervised anomaly detection based on clustering and multiple one-class SVM IEICE Trans. Commun. E92-B 1981–90

[100] Gyanchandani M, Yadav R N and Rana J L 2010 Intrusion detection using C4. 5: performance enhancement by classifier combination Proceedings of the International Conference on Advances in Computer Science pp 130–3

[101] Gudadhe M, Prasad P and Kapil Wankhade L 2010 A new data mining based network Intrusion Detection model 2010 International Conference on Computer and Communication Technology (ICCCCT) pp 731–5

[102] Raj Kumar P A and Selvakumar S 2011 Distributed denial of service attack detection using an ensemble of neural classifier Comput. Commun. 34 1328–41

[103] Bahri E, Harbi N and Huu H N 2011 Approach Based Ensemble Methods for Better and Faster Intrusion Detection Computational Intelligence in Security for Information Systems ed Á Herrero and E Corchado (Berlin, Heidelberg: Springer Berlin Heidelberg) pp 17–24

[104] Lin L, Zuo R, Yang S and Zhang Z 2012 SVM ensemble for anomaly detection based on rotation forest 2012 Third International Conference on Intelligent Control and Information Processing pp 150–3

[105] Balon-Perin A and Gambäck B 2013 Ensembles of decision trees for network intrusion detection systems Int. J. Adv. Secur. 6

[106] Masarat S, Taheri H and Sharifian S 2014 A novel framework, based on fuzzy ensemble of classifiers for intrusion detection systems Proc. 4th Int. Conf. Comput. Knowl. Eng. ICCKE 2014 165–70

[107] Gaikwad D P and Thool R C 2015 Intrusion detection system using Bagging with Partial Decision Tree base classifier Procedia Comput. Sci. 49 92–8

[108] Amini M, Rezaeenour J and Hadavandi E 2016 A neural network ensemble classifier for effective intrusion detection using fuzzy clustering and radial basis function networks Int. J. Artif. Intell. Tools 25 1550033

[109] Pham N T, Foo E, Suriadi S, Jeffrey H and Lahza H F M 2018 Improving performance of intrusion detection system using ensemble methods and feature selection ACM International Conference Proceeding Series p 2

[110] Vinutha H P and Poornima B 2018 An ensemble classifier approach on different feature
selection methods for intrusion detection Advances in Intelligent Systems and Computing vol 672 (Springer) pp 442–51

[111] Chou T-S S, Fan J, Fan S and Makki K 2009 Ensemble of machine learning algorithms for intrusion detection Conf. Proc. - IEEE Int. Conf. Syst. Man Cybern. 3976–80

[112] Zainal A, Maarof M A, Shamsuddin S M and others 2009 Ensemble classifiers for network intrusion detection system J. Inf. Assur. Secur. 4 217–25

[113] Liu G, Chen W and Hu F 2010 A neural network ensemble based method for detecting computer virus 2010 Int. Conf. Comput. Mechatron., Control Electron. Eng. C. 2010 1 391–3

[114] Govindarajan M and Chandrasekaran R 2011 Intrusion detection using neural based hybrid classification methods Comput. Networks 55 1662–71

[115] Boro D, Nongpoh B and Bhattacharyya D K 2012 Anomaly based intrusion detection using meta ensemble classifier Proc. 5th Int. Conf. Secur. Inf. Networks, SIN’12 143–7

[116] Govindarajan M and Chandrasekaran R M 2012 Intrusion detection using an ensemble of classification methods Proc. of the World Congress on Engineering and Computer Science vol 1 pp 459–64

[117] Meng Y and Kwok L-F 2013 Enhancing False Alarm Reduction Using Voted Ensemble Selection in Intrusion Detection Int. J. Comput. Intell. Syst. 6 626–38

[118] Chaurasia S and Jain A 2014 Ensemble neural network and k-NN classifiers for intrusion detection Int. J. Comput. Sci. Inf. Technol. 5 2481–5

[119] Govindarajan M 2014 Hybrid intrusion detection using ensemble of classification methods Int. J. Comput. Netw. Inf. Secur. 6 45–53

[120] Tama B A and Rhee K H 2015 A combination of PSO-based feature selection and tree-based classifiers ensemble for intrusion detection systems In Advances in Computer Science and Ubiquitous Computing vol 373 (Springer) pp 489–95

[121] Jabbar M A, Srinivas K and Sai Satyanarayana Reddy S 2018 A novel intelligent ensemble classifier for network intrusion detection system Advances in Intelligent Systems and Computing vol 614, ed A Abraham, A K Cherukuri, A M Madureira and A K Muda (Cham: Springer International Publishing) pp 490–7

[122] Jabbar M A, Aluvalu R and Reddy S S 2017 RFAODE: A Novel Ensemble Intrusion Detection System Procedia Comput. Sci. 115 226–34

[123] Rajasekaran M and Ayyasamy A 2017 A novel ensemble approach for effective intrusion detection system Procedings - 2017 2nd International Conference on Recent Trends and Challenges in Computational Models, ICRTCCM 2017 pp 244–50

[124] Salo F, Nassif A B and Essex A 2019 Dimensionality reduction with IG-PCA and ensemble classifier for network intrusion detection Comput. Networks 148 164–75

[125] Ludwig S A 2019 Applying a Neural Network Ensemble to Intrusion Detection J. Artif. Intell. Soft Comput. Res. 9

[126] Zhou Y, Cheng G, Jiang S and Dai M 2020 Building an efficient intrusion detection system based on feature selection and ensemble classifier Comput. Networks 174 107247

[127] Tama B A, Nkenyereye L, Islam S M R and Kwak K 2020 An Enhanced Anomaly Detection in Web Traffic Using a Stack of Classifier Ensemble IEEE Access 8 24120–34

[128] De La Hoz E, Ortiz A, Ortega J and De La Hoz E 2013 Network anomaly classification by support vector classifiers ensemble and non-linear projection techniques Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics) 8073 LNAI 103–11

[129] Timčenko V and Gajin S 2017 Ensemble classifiers for supervised anomaly based network intrusion detection Proceedings - 2017 IEEE 13th International Conference on Intelligent Computer Communication and Processing, ICCP 2017 pp 13–9

[130] Mousavi S M, Majidnezhad V and Naghipour A 2019 A new intelligent intrusion detector based on ensemble of decision trees J. Ambient Intell. Humaniz. Comput.

[131] Gu J, Wang L, Wang H and Wang S 2019 A novel approach to intrusion detection using SVM ensemble with feature augmentation Comput. Secur. 86 53–62

[132] Bhati B S and Rai C S 2020 Ensemble Based Approach for Intrusion Detection Using Extra
Tree Classifier *Intelligent Computing in Engineering* ed V K Solanki, M K Hoang, Z (Joan) Lu and P K Pattnaik (Singapore: Springer Singapore) pp 213–20

[133] Manju N and Harish B S 2020 Classification of Internet Traffic Data Using Ensemble Method *Progress in Computing, Analytics and Networking* ed H Das, P K Pattnaik, S S Rautaray and K-C Li (Singapore: Springer Singapore) pp 397–405

[134] Yan J, Yun X, Zhang P, Tan J and Guo L 2010 A New Weighted Ensemble Model for Detecting DoS Attack Streams *2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology* vol 3 pp 227–30

[135] Haq N F, Onik A R and Shah F M 2015 An ensemble framework of anomaly detection using hybridized feature selection approach (HFSA) *2015 SAI Intelligent Systems Conference (IntelliSys)* pp 989–95

[136] Elekar K S 2016 Combination of data mining techniques for intrusion detection system *IEEE Int. Conf. Comput. Commun. Control, IC4 2015* 1–5

[137] Rajagopal S, Kundapur P P and Hareesha K S 2020 A Stacking Ensemble for Network Intrusion Detection Using Heterogeneous Datasets ed S Cimato *Secur. Commun. Networks 2020* 4586875

[138] A. J. Obaid, K. A. Alghurabi, S. A. K. Albermany and S. Sharma, "Improving Extreme Learning Machine Accuracy Utilizing Genetic Algorithm for Intrusion Detection Purposes," in Advances in Intelligent Systems and Computing, Springer, Singapore, 2021, pp. 171-177.