Science in Information Technology Letters
Vol. 2., No. 2, November 2021, pp. 1-14
ISSN 2722-4139
http://pubs2.ascee.org/index.php/sitech

A Fundamental Overview of SOTA-Ensemble Learning Methods for Deep Learning: A Systematic Literature Review

Marco Klaiber [a,1,*]

[a] Aalen University of Applied Sciences, 73430 Aalen, Germany
[1] marco.klaiber@studmail.htw-aalen.de
* Corresponding Author

ARTICLE INFO

Article history
Received Sept 12, 2021
Revised Oct 20, 2021
Accepted Nov 15, 2021

Keywords
Ensemble Learning
Bagging
Boosting
Deep Learning
Machine Learning
Predictive Performance
CNN

ABSTRACT

The rapid growth in popularity of Deep Learning (DL) continues to bring more use cases and opportunities, with methods rapidly evolving and new fields developing from the convergence of different algorithms. For this systematic literature review, we considered the most relevant peer-reviewed journals and conference papers on the state of the art of various Ensemble Learning (EL) methods for application in DL, which are also expected to give rise to new ones in combination. The EL methods relevant to this work are described in detail and the respective popular combination strategies as well as the individual tuning and averaging procedures are presented. A comprehensive overview of the various limitations of EL is then provided, culminating in the final formulation of research gaps for future scholarly work on the results, which is the goal of this thesis. This work fills the research gap for upcoming work in EL for by proving in detail and making accessible the fundamental properties of the chosen methods, which will further deepen the understanding of the complex topic in the future and, following the maxim of ensemble learning, should enable better results through an ensemble of knowledge in the future.

1. Introduction

In recent years, Deep Learning (DL) has become more and more relevant, and its uses are constantly increasing [1]–[3]. DL is often equated with Machine Learning (ML), but it should be noted that DL is a subset of ML, and both belong to the category of Artificial Intelligence (AI) [4], [5]. ML is defined as an independent, evolving branch of computing algorithms that aims to replicate human intelligence by learning from the environment [6]. DL, on the other hand, structures algorithms into layers to create an artificial neural network (ANN) that can learn autonomously and make intelligent decisions [1], [3], [7]–[9]. A neural network uses the knowledge it has learned during training and applies it to unseen data to best solve a given problem, which is called the generalization ability of a neural network [10], [11]. The monumental characteristic of DL is the ability of the layers to learn the individual features from the data themselves, using a universal learning method, which makes it suitable for numerous applications in many domains of science, economy, and administration [12].

A significant problem with traditional DL, i.e., training and assessment of a single model, is that the results are based on a single model, which may have limited and therefore unsatisfactory outcomes, which can be affected by overfitting, or generally poor performance due to over-complex data, unbalanced datasets, incorrect hyper-parameters, etc. [13]. The reasons for poor performance can be enormously multifaceted, so relying on a single model entails risks, as a model must capture multiple complex features alone, which can be a significant difficulty and can lead to its uselessness [13]. This is exactly where Ensemble Learning (EL) comes into play. EL is a way to solve DL problems by combining several individual models that provide independent results and aggregating them to make a final decision based on consensus [1], [9], [10], [14]. EL is thus an effective way to optimize generalization, predictive performance, and robustness through a combined model [1], [10], [11], [14]–[16].

According to Dietterich [17], there are 3 different reasons why an ensemble can perform better than a single learner. Firstly, there is a statistical reason that the training data does not contain enough information and therefore it is not possible to identify the best learner. There may be several individual learners who present identical results, whereby the combination of the individual learners may be a better choice. The second reason is that the search process of the learners may be imperfect. The implication is that even when there is clearly the best hypothesis, it may be difficult to achieve it because the execution of individual learners may lead to suboptimal hypotheses. This is where EL steps in, as it is considered a way to compensate for an imperfect search process. As a final reason Dietterich mentions that the hypothesis space sought may not contain the true objective function, whereas an ensemble could provide a good approximation. Dietterich nevertheless points out that these reasons are intuitive and not strict theoretical rules. Three years later Dietterich consolidated all these reasons and gave each of them an umbrella term: the statistical, computational, and representational aspect.

The use of an ensemble of different DL models has, in addition to the advantages mentioned, potential complications such as interpretability, training complexity and high hardware performance requirements, which are addressed by the innovative approaches also presented in this research. Because of this, it is very important to have a deep understanding of the most popular EL techniques and how they work, to further improve DL models, which is of great importance for a more extensive adaptation in real world applications. In addition, there are various methods of EL in the domain of DL, but an in-depth up-to-date overview of the favored ways to form an ensemble is missing to the best of my knowledge. Exactly this gap will be filled by this work to provide a basis for future work in connection with EL, to build up and develop further work with profound knowledge. Furthermore, the work is intended to provide an up-to-date state of research on the most frequently used methods of EL for use in DL, to promote innovative approaches and – according to the maxim of EL – to achieve better results in the future through an ensemble of knowledge.

This paper addresses the current state of research on various EL methods for use in DL, focusing not only on the fundamental properties of the most popular EL methods, but also on uncovering further optimization potential. At the beginning, a general overview of the field of DL is given. It will be shown what EL is needed for and which methods are used, with the most popular ones resulting from the research. Then, the individual EL methods relevant to this work are pointed out and described in detail. The result of this work is to present the individual limitations and optimization possibilities of EL methods, which future research in this area should systematically consider and address in order to further advance EL research.

2. Method

To filter out the relevant results from the published literature, a systematic literature research with the search word combination "Ensemble Learning" and "Deep Learning" was carried out in the literature databases IEEEXplore DL and ScienceDirect. The search resulted in a total of 93 articles that contain the mentioned search word combination in the title, abstract or keywords. This was supplemented by a forward and backward search to cover the fundamental information of the research topic and build a citation network. To ensure that only significant scientific papers are included in this search, only peer-reviewed journals and conference papers were considered. After manually reviewing these 93 articles, 51 articles were classified as relevant for the object of investigation set up at the beginning.

2.1. Ensemble Learning

The idea of EL can be reflected based on a real-life situation. When a company is faced with a critical decision, several opinions of experts are consulted instead of relying on a single judgement [18]. The EL is built on the same principle: a common decision is made based on several individual decisions of models [9], [18], [19]. In the context of DL, an ensemble can be defined as a ML system composed of several individual models that learn in parallel or sequentially [1], [9], [10], [14], [17], [19]–[21]. The individual results are combined to find a collective solution to a specific problem [1], [9], [10], [14], [17], [19]–[21]. Basically, EL combines different models, also called classifiers, to achieve better performance than a single classifier [3], [14], [15], [21]–[23].

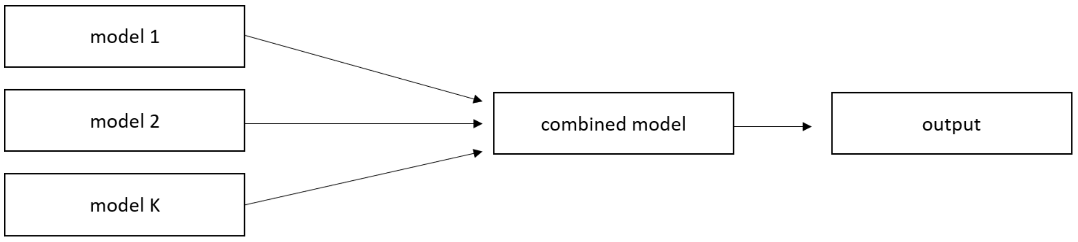

Fig. 1. General presentation of Ensemble Learning; in accordance with [10].

Within EL, there is talk of “collectivism”, since the individual classifiers learn independently of each other, but nevertheless work out a common solution [9]. The individual classifiers are also referred to as learners, and the concept of the learner is further divided into weak and strong learners [9], [10]. A weak learner, also called a basic learner, is a classifier within an ensemble. A strong learner is defined by the combination of the individual weak learners or the individual classifiers [10], [18]. EL can create a strong learner from several weak learners, each of whom must be slightly better than random assumptions, and can make accurate predictions [9], [10], [24], [25]. The individual learners can be similar in type, so a homogeneous integration occurs, i.e., the individual classifiers are the same. When learners are diverse, it is called heterogeneous integration [9].

There are two core factors that define “good” EL: the accuracy of the individual classifiers and the diversity of the classifiers [1], [3], [8], [10], [19], [23], [26]. The diversity can be further subdivided into data, parameter, and structural diversity [8]. Data diversity describes the individual datasets that are made available to the individual classifiers, which should be as different as possible from each other. Parameter diversity covers the variation of the classifiers’ parameters, e.g., the weights and activation functions. Structural diversity includes the different architectures and structures, i.e., the structure of each classifier [8], [19]. The diversity is of enormous importance because it is directly correlated with the performance of an ensemble [1], [8], [10], [19], [23].

A significant point of EL is the combination strategy, which is elementary for ensemble performance [11], [15]. The combination strategy is used to combine the different models and their results to derive an overall result [18]. It is important to understand in advance that there is no one best combination strategy, but that the performance of an ensemble depends on the domain and the problem [17], [18]. It can be stated that an ideal ensemble consists of classifiers with high accuracy that differ as much as possible [18]. The reason for this is that classifiers with different errors reduce the total error. If individual classifiers make identical mistakes, an ensemble will appear useless because the opinions of weak learners agree [18].

2.2. Combination Strategies

EL algorithms, also called combination strategies, are so-called meta-algorithms that combine different models to create a single prediction model [3], [27]. It is important to understand that there is no combination strategy that consistently outperforms other combination strategies [10], [17]. Each combination strategy has specific advantages and aspirations that should be achieved by a combination. A combination should optimize the predictive performance, the ability to generalize, as well as the robustness of an ensemble. Therefore, it is relevant to know the individual combination strategies and their specifics to use them efficiently [28]. The ability to generalize includes a further goal of EL, namely the reduction of possible overfitting within the ensemble model [29]–[31]. Overfitting implies that during the training phase, the model learns the features and specifics of the training dataset extremely well but performs poorly within the test data or on data never seen before [10], [32]. Generalizability would be poor and unreliable, although EL can reduce the possibility of overfitting [10], [32].

In the research, the following combination strategies turned out to be most relevant: ”Boosting”, including the extended form ”AdaBoost” [33]–[36], the bootstrap aggregation known as ”Bagging” [37], ”Wagging” and ”Stacking”, which follow modified and extended approaches, respectively [10], [18], [23] and ”Random Forests”, as well as the modern ”Snapshot” procedure [8], [11], [18], [22], [38].

2.2.1. Boosting

Boosting is a concept that combines weak classifiers to form a strong classifier with high classification power and generalization capability. In 1990, Schapire presented Boosting for the first time to show that if the classification performance of a weak classifier were higher than 0.5, it would be possible to combine several weak classifiers into one strong classifier [9], [27], [39]. The logic of boosting is that it is much more efficient to find several small rules of thumb for a problem than to find one general rule that solves a particular problem [39]. The procedure is as follows: In an ensemble, the first learner is formed and trained on the entire training dataset; this first learner is only effective in identifying the data [9], [23]. Subsequently, further ensemble members are formed incrementally, who are assigned the adapted training data based on the previous learner. Adapted implies that a greater focus is placed on misclassified data, thus focusing on the errors of the previous ensemble members. In addition, complex and difficult-to-identify data is given a higher sampling probability in this process. The next iteration focuses on the previously distributed training data. This process is repeated until a predefined number of iterations is reached [9], [23]. The idea behind this is to execute the weak learners several times with different distributed instances to obtain several classification results [39]. Finally, the classifications of all trained learners are aggregated, and the correct classification is determined by a voting procedure. The vote of a classifier with a lower accuracy is weighted less than the vote of a classifier with a high accuracy [23].

Boosting nevertheless shows difficulties. On the one hand it is necessary to know the minimum learning performance of a weak learner, this is difficult to do in reality [9]. On the other hand, boosting can lead to an excessive focus on data that is difficult to train, which can result in a poorer performance. In addition, the training record should be adapted so that weak learners can learn the information that is not included in future iterations. Another major limitation of boosting is that it can be applied primarily to binary classification problems, since it can assume a result of 1 or -1, so that multiclass problems cannot be solved, or rules for them are lacking [18]. In addition, boosting is particularly sensitive to outliers and noisy data, which can degrade performance depending on the dataset [9], [27].

Nowadays an extended form of boosting is usually used, namely adaptive boosting (AdaBoost) [33]–[36]. AdaBoost was presented by Freund and Schapire in 1997 and is still the most representative and well-known algorithm within the family of boosting algorithms [34]. AdaBoost builds directly on the Boosting algorithm presented by Schapire and extends it [34]. The adaptive approach defines that the algorithm can adapt the weak learners after the training [9]. As with the predecessor, an ensemble is built incrementally, with each newly added ensemble member being trained to highlight the previously misclassified training instances [33], [35]. In AdaBoost, the individual weak learners must also be somewhat better than pure chance [18], [36]. Compared to the original boosting, the random sample is replaced by weighted samples within the training. This implies that the random sample is replaced by a weighted sample. This change allows the focus to be placed on training instances that are difficult to process, making the training of a weak learner much more targeted [9]. AdaBoost places a lot of emphasis on data that has been incorrectly classified and weights the correctly classified data less, which is an effective way of enabling the later strong classifier to make the right decision effectively [33], [36].An advantage of AdaBoost is that it is not necessary to know the learning efficiency of a weak learner in advance, since the classification performance of the strong classifier is directly dependent on the classification performance of the weak classifiers, and it is therefore possible to draw conclusions about the individual integrated weak classifiers. AdaBoost is an efficient method to improve the generalization capability of the resulting strong classifier and to reduce errors within a classification problem and solve the problem efficiently. This effective error reduction is attributed to adaptive re-sampling by Breiman, who suggested the following "Bagging". Furthermore, boosting is not very susceptible to overfitting, but is sensitive to noise data and outliers [40], [41]. Boosting is popular because it adapts the training data epochally, and selectively combines them depending on the individual weights of the classifiers, rather than using specific fusion rules [35]. In addition, the algorithm effectively controls the individual integrated weak classifiers, which leads to an increase in execution efficiency, thus enabling real-time application.

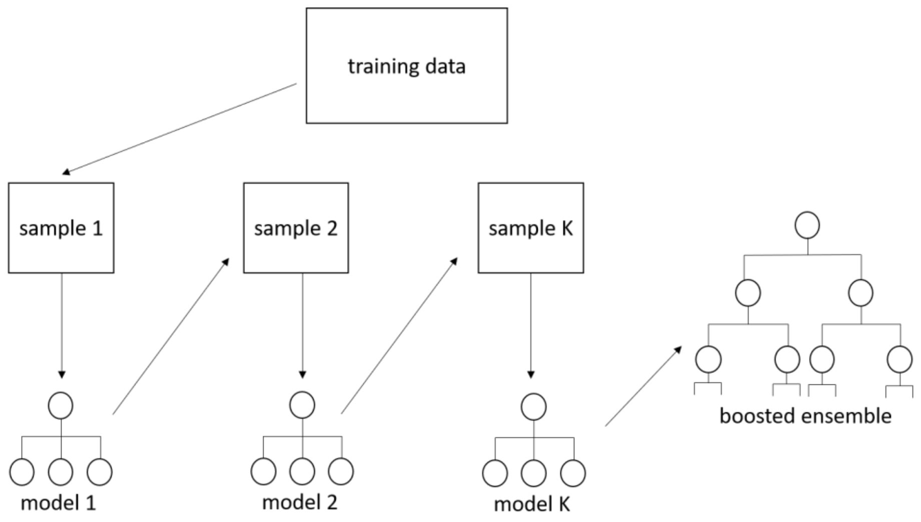

Fig. 2. Boosting diagram; in accordance with [21].

2.2.2. Bagging

The acronym Bagging stands for "Bootstrap Aggregation", which was presented by Breiman in 1996 [20]. It is one of the most intuitive and earliest EL-algorithms, yet with a good performance [18]. Bagging is described in the literature as the most well-known form of EL, with the random forest algorithm as the poster child for bagging, which is also described in detail in this paper [14].

The bagging algorithm provides the ability to effectively solve classification and regression problems, and the bootstrap algorithm proposed by Tibshirani and Efron [42] occupies an elementary role. Bootstrap generates random replications of the training dataset that are assigned to the individual models within the ensemble [42]. Each training dataset is generated from N instances, including replacements, from the entire dataset M, with uniformly random selection [18], [19], [37]. The individual bootstrap replicas contain an average of 63.2% of the original total training dataset, with some instances occurring more than once [18], [33]. Bootstrap replications increase the diversity of classifiers, which is of immense importance [18], [19]. The individual bootstrap samples are made available to the classifiers for training, whereby each training dataset is different from the others [18]. Significantly, each of the classifier's trains completely independently of the others and thus the prediction results of each model are calculated independently and in parallel [3], [18], [21], [37]. Then the individual results are aggregated and combined [3], [18], [37]. By combining the results, a final decision can be made, which determines the decision of the ensemble [18]. The decision is combined by different methods, the most relevant being the sum and majority vote [21]. EL methods work best with unstable classifiers, including bagging [18], [37]. Unstable classifiers have different generalization capabilities, which are also called high variance classifiers [18]. Bagging can therefore be a way to exploit the instability of classifiers and thus optimize the prediction accuracy. Due to this effectiveness with unstable models, bagging is often used in the domain of DL using neural networks that are considered unstable due to the random initialization of the individual weights. Bagging works less well with extremely simple classifiers that have a high probability of making identical predictions, which leads to low diversity and poor result. Effective is the random replication of the training data on the same cardinality of the original training data, so the size of the sample is identical to the size of the training dataset. Bagging mainly aims at reducing the variance within the predictions of the classifiers to achieve a better result than single classifiers [18].

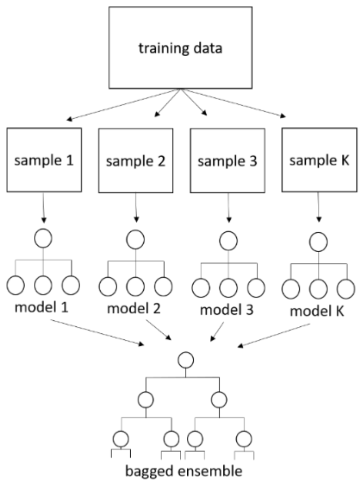

Fig. 3. Bagging diagram; in accordance with [15] [28].

2.2.3. Wagging

Wagging, which was proposed by Bauer and Kohavi [40] in 1999, is an extended form of bagging and is therefore called "Weighted Bagging" [43]–[45]. Wagging is like bagging except that each weak learner is trained on the entire training dataset, with each instance being assigned a weight stochastically [43]–[45]. Instead of using random bootstrap samples, weights are randomly assigned to form the successive training datasets, whereby the number of total samples is kept [45]. Originally, Bauer Kohavi used "Gaussian Noise" to vary the individual weights of the instances. This occasionally resulted in some instance weights being reduced to 0, removing instances from the training dataset. As a result, Wagging was adapted at the suggestion of Quinlan, who implemented the Poisson distribution, known as exponential distribution.

As in bagging, each classifier learns independently from the others, which also enables a fast and parallel process [45]. The variation of the data is achieved by adjusting the weights assigned to the instances, thus not by the data instances themselves. Wagging now follows the same process as bagging, because the results of the individual weak learners trained are combined, which leads to a final classification [43]–[45].Weighted bagging offers an interesting possibility to change the influence of each sample on the weak learner, within the training, using the weights. Wagging is less effective than bagging in reducing errors due to the inclusion of each instance of the training dataset, resulting in less variation between the resulting ensemble members [23]. Nevertheless, Wagging is often preferred to Bagging because it interacts better with learning algorithms, which can be explained by the full integration of all training cases.

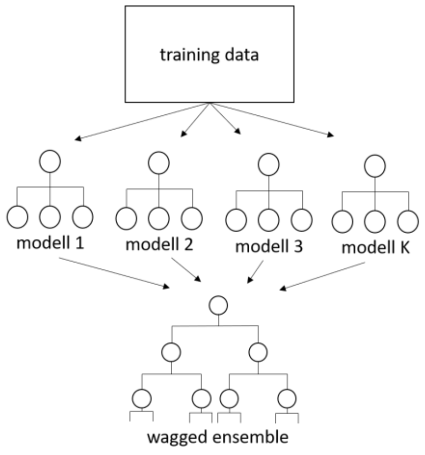

Fig. 4. Wagging diagram; in accordance with [15].

2.2.4. Stacking

Stacking or "Stacked Generalization" was proposed by Wolpert in 1992 [46]. Stacking is primarily like boosting, but unlike boosting and bagging, stacking is often used to combine different types of models [18], [31]. Stacking ensures the variation and generalization of the resulting ensemble through the different models, the basic idea being to identify training data that has not been learned correctly [18], [31]. During stacking, the entire training dataset is divided into M-blocks, with a distinction being made between training and test sets within the M-blocks [31]. In the first instance, the entire training dataset is divided into training and test sets, and then the training datasets are further divided using "cross validation" [28]. The cross-validation has the peculiarity of stacking that a combination takes place subsequently instead of the "winner-takes-it-all" approach [18], [31]. Cross-validation is used to increase computational efficiency and to avoid possible overfitting [31]. In the first place, the weak learners are trained. When the whole dataset is split up, a pseudo test is retained, so that the weak learners can be tested on data not yet seen after the training. There are different types of classifiers for stacking, or classifiers divided into levels. On the one hand, there are the level-0 classifiers, which provide the input for learning a level-1 classifier through their output. The output of level 0 thus becomes the training dataset of level 1, whereas the level 1 classifier is also called meta classifier. This is helpful in case a classifier in level 0 has learned a certain domain incorrectly and thus consistently classifies the instances incorrectly. A level 1 classifier can possibly learn this behavior together with the individual behavior of other weak learners [18], [27]. Thus, the level 1 classifier can correct the wrong classification. When combining the level 0 classifiers to a level 1 classifier it is relevant that the algorithms have a low correlation [27]. This allows the meta-classifier to determine the optimal performance of the model [27].

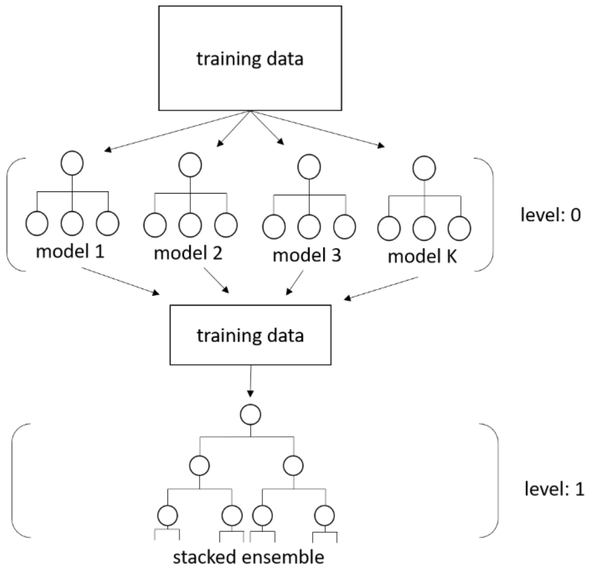

Fig. 5. Stacking diagram; in accordance with [15], [35].

2.2.5. Random Forest

The The Random Forest (RF) is an EL method with a combination of many decision trees, which is mainly used for classification, regression, and possible other tasks. The RF is a variation of bagging, which it combines the concepts of bagging and "random subspaces" and merges them [47]. Bagging works with arbitrary models as weak learners, whereas RFs are ensembles of undivided classification or regression trees, thus distinguishing them from other DL models [18]. To train the datasets, several decision trees are combined into one RF as weak learners [8], [18]. Each tree within an RF is grown according to a random tree-building scheme and without pruning [48]. The variation of the individual decision trees depends directly on the randomly generated tree structure, resulting in different classifications [48]. For example, in classification, each individual tree in the forest decides. Then the votes of the trees are counted and the class with the most votes, in case of a majority vote, wins the vote and the problem is solved based on the vote [18], [29], [30].

This EL method is characterized by the fact that it can train datasets relatively quickly and simply, since the trees make their decisions independently of each other, thus enabling parallel processing [18], [29]. Despite the parallelism, Breiman pointed out that AdaBoost has some similarities to a RF, although AdaBoost functions incrementally [8], [29]. The RF, compared to boosting, is still more robust and faster in the training process. This is due to the stable weighting and division of the data into small subsets. An RF acts efficiently regarding large amounts of data, and it is possible to process thousands of variables without having to delete variables [18]. In addition, RF are relatively easy to extend to an online version [18].

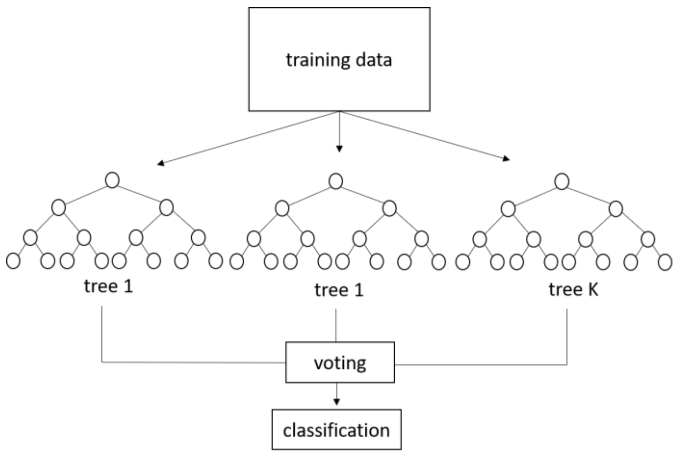

Fig. 6. Random Forest diagram; in accordance with [9], [15].

2.2.6. Snapshot Ensemble Learning

The Snapshot Ensemble Learning (SEL) was introduced by Huang et al. [38] in 2017. This method is the most innovative of all presented methods and addresses one of the biggest problems of EL: to minimize the enormous computing effort of an ensemble [9], [11], [22], [38].

The SEL promises the performance of an ensemble at the expense of a single network by continuously taking snapshots in local minima of the search area of a network, which are finally merged [11], [22], [38]. The underlying idea is to have an ensemble of different and accurate models emerge from a single training process [22], [38]. Diverse and precise refers to the fact that the individual models have a small test error and do not overlap within their incorrect classifications [38]. At the heart of SEL is the optimization process, which seeks out multiple local minima before converging to a final solution [38]. The SEL process is divided into a cyclical learning rate plan. This implies that the entire training process is divided into M-cycles, which results in M-various network models [11], [22], [38]. At the end of the training cycles, the local minima in terms of training loss become apparent, which in practice is achieved for example by the stochastic gradient descent (SGD) method with warm restarts [22], [38]. To achieve multiple local minima, a cyclical annealing schedule presented by Loshchilov and Hutter [49] in 2016 is used. At each arrived minimum, a snapshot of the model weights is taken, then the learning rate is further adjusted [22], [38]. The learning rate is first decreased rapidly to promote rapid convergence, and then increased once, moving the model away from a local minimum [22]. This process is repeated for the next local minima, which allows the cost of a single training session to be maintained [22]. Of relevance is the fact that the training time of the cyclical learning rate plan hardly differs from a normal learning rate plan [38]. After training the M-cycles, M-model snapshots are achieved, which are used in the final ensemble [38]. The M-local minima allow increasingly accurate predictions to be made during training [38]. The individual snapshots are accurate and provide different predictions and are therefore well suited for an ensemble that can make a valid decision by combining them [11].

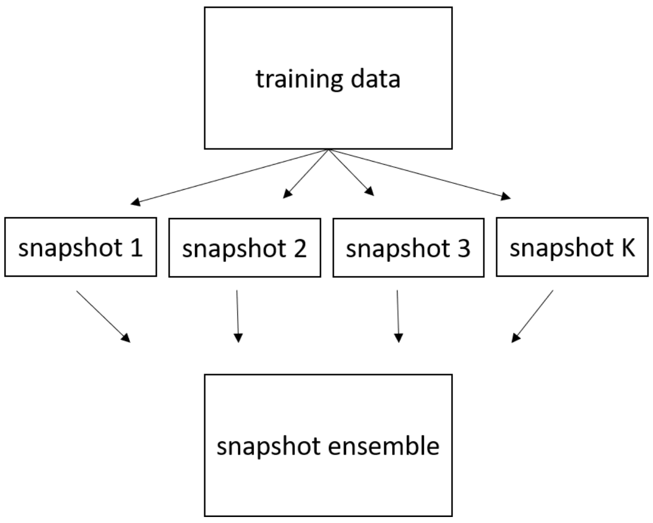

Fig. 7. Snapshot Ensemble diagram; in accordance with [20], [29].

2.3. Voting Procedures

Within EL, the various weak learners are trained and their results, and thus the output, are combined [18]. To make the right decision based on the different results, different voting and averaging procedures are used to determine how the solution to the problem should be classified. The voting process is necessary to make a final decision [3]. [36] generalize the individual procedures and presented that majority voting is primarily used for classification tasks. In contrast, the normal and weighted average procedure is primarily used for regression tasks [36]. Two different types of voting algorithms are distinguished: those that adaptively change the distribution of the training set based on the performance of previous classifiers, as in boosting, and those that do not, as in bagging, or RF [40].

2.3.1. Sum-method

This method selects the class that has the highest sum of all probabilities [21]. All results of classifiers predicting the same result are added to form a sum of the respective class. The final decision is made on the class with the highest summed probability [21].

2.3.2. Majority-vote

The majority vote follows the concept of selecting the class that is most frequently named or voted for [50]. Each classifier delivers a vote for a class, which is indicated by the highest integrated result [19], [25]. Furthermore, it is possible to add weights to the results of each classifier to decide [21]. This scheme is most often used in classification applications [10].

2.3.3. Unweighted-average

In this method, the individual outputs of the different classifiers are averaged [17], [19], [25]. The calculation of the average is calculated with an equal weighting for all classifiers, as is the case for example with bagging [17], [25]. The output that has the maximum of the averaged values is selected as the correct class and is applied to the problem [19], [25]. This method can only be used if the outputs of the individual classifiers are numerical [19]. According to [25], the unweighted average works particularly well if the different classifiers are similar.

2.3.4. Weighted-average

This method weights the individual outputs of the classifiers, whereby the outputs are also averaged [19], [25], [50]. A certain set of weights must be found that reflects the individual performances of the classifiers [25]. The goal of the weightings is to minimize the difference between the ensemble's performance and the desired output [19]. An error correlation matrix can be used to determine the weights [19]. This method is particularly popular for AdaBoost and regression applications [9], [10].

3. Results and Discussion

3.1. Predictive performance and limitation of EL

A learner's suitability for a particular domain is usually assessed based on predictive performance, which indicates how well the model will perform [28]. Predictive performance plays a crucial role in EL since an ensemble of several models should normally perform better than a single model [8]. Basically, an ensemble consists of several combined models. This combined ensemble model should outperform every single model within the ensemble [8], [9].

Lee et al. [9] showed that an ensemble can effectively compensate for the probability that an individual learner is misclassified. It also addresses the fact that combining several learners reduces the risk that individual learners will perform equally well and thus have a poor ability to generalize [9]. In addition, an ensemble reduces the risk of possible overfitting in terms of training data [9], [10]. Individual learners tend to capture a partial optimum, but the generalization ability of some partial optima can be poor, this risk can be reduced by integrating several learners [9]. This last point overlaps with the findings of Dietterich [17] from 1997, because an ensemble can expand the hypothesis space of individual learners, which improves the learning of an approximation function [9].

Other works also speak of an increase in the ability to generalize, using EL. Wang et al. [14] showed, for example, that the ability to generalize is usually much better than that of a single learner when several learners are combined [14]. In addition, they speculated that as the amount of data increases, the proposed EL method will also improve performance [14]. In the domain of image classification, Chen et al. [1] showed that an ensemble has a high potential and better classification accuracy than a single classifier. Nevertheless, they made it clear that an ensemble requires a much longer processing time [1]. Ren et al. [8]presented a performance enhancement by EL, especially in deep neural networks the performance can be improved.

Webb and Zheng [23] focused explicitly on boosting and bagging and presented equally impressive improvements in the domain of prediction accuracy and generalization capability. Thus, it can be concluded that EL improves not only predictive performance but also generalization capability, accuracy, and robustness by combining several classifiers [51]. The list of successful integrations of EL will grow in the future, however, there are limitations and weaknesses within EL that limit the use of an ensemble [1], [8], [9], [17], [21], [22]. In particular, the required computing capacity, interpretability, and training complexity play a major role [1].

The main disadvantage of an ensemble is the enormous computational effort, because K different models are trained instead of a single model [22]. An ensemble requires an enormous amount of computing power [1], [22]. On the one hand, the data must be saved and on the other hand many calculations must be performed [17]. At that time, Dietterich had given the example that an ensemble with 200 decision trees achieves a perfect performance within one classification. The 200 decision structures required about 59 megabytes of memory, which was a lot in 1997 [17]. Due to the progress in technology, a much larger amount of data can be processed today, but the amount of data is still growing, especially in the age of Big Data, so the problem of computing capacity is still one of the most relevant [1], [9], [22], [38]. Explicitly within deep networks the EL is enormously computationally intensive [9], [22], [38]. From a computational point of view, deep networks are known for their high training costs, so that combinations are only feasible with the necessary computing capacity [9], [22], [38].

In terms of interpretability, EL still has potential for improvement [10], [17], [19]. The various ensemble methods lack a clear understanding of the underlying process [10]. In some cases, theoretical explanations of phenomena that occur when using EL are missing [10]. In addition, the learned knowledge of an ensemble is not easily understood by the user, which can cause problems for the user [19]. The user gains little insight into the decision-making process, so a single decision tree can be interpreted by one person, but to interpret a RF of K-decision trees is nearly impossible [17].

Training complexity is another major difficulty within EL [10], [17]. First and foremost, the most effective combination strategy must be chosen when using EL. This presents a difficulty because, as mentioned above, there is not one combination strategy that consistently outperforms all others [10] [17]. Thus, there cannot be the very best EL model for all applications, since the respective performance depends strongly on the chosen domain and the specific requirements [10], [17]. If the data is too complex, it can lead to the fact that the weak learners all cannot be trained sufficiently and, moreover, they are very similar to each other. The enormous importance of the variation between the individual classifiers was often emphasized in this work and correlates directly with the training complexity [1], [3], [8], [10], [19], [23]. If there is no diversity within an ensemble, the members of the ensemble will learn the same mistakes during training and thus produce an unsatisfactory result as a combined ensemble [1], [3], [8], [10], [19], [23].

The individual diversity measures are not yet pronounced, which makes diversity and thus training more difficult [19]. Within the training of an ensemble, the choice of the threshold value for each weak learner is also important, because if a value is not correctly defined, it can increase the susceptibility to errors within the model [25]. When training data changes, outliers, incorrect samples, or noise can cause bias, resulting in the so-called drift problem [24]. This can cause the EL model to gradually lose focus on the true target, a phenomenon that occurs primarily in real-time applications [24].

3.2. Limitations & Future Work

The limitation in this study was the transparency of the works to be compared, since not all of them provided the required information in full. In addition, it was limited to the selected essential procedures and models, which means that possible further and innovative procedures must be analyzed in future works, besides the most frequently used ones from this work.

In the domain of EL models for use in DL, the future focus will be on increasing efficiency in terms of interpretability, reducing training complexity, and lowering hardware performance requirements. When working with EL, an improvement in the results, compared to a single model, can be assumed. However, the individual EL methods take different approaches and therefore produce different results. As a result, it is often unclear in advance which method will yield the best results, and the optimal solution must be sought through time-consuming experimentation with different methods and parameters [26]. Here, time saving potentials regarding the increase of efficiency in the selection of the appropriate methods can be identified. In the future the focus should also be on interpretability. Often the user receives a good result, but still has little insight into the decision making of the ensemble [10], [17], [19]. The internal structures of an ensemble can hardly be explained clearly, this is where the focus should be [10], [17], [19]. In general, interpretability seems to be a problem not yet strongly addressed by EL, although it could further advance the progress of EL [10]. The methods of EL in the domain of DL presented in this literature review often have the problem that they place high demands on hardware resources and are therefore not suitable for use by the public [1], [9], [22], [38]. One reason for this is that instead of a model, K models are trained to deliver the desired results [22]. In the future it will be a task to design applications more resource efficient without losing accuracy and efficiency.

4. Conclusion

All major peer-reviewed journals and conference papers – supplemented by a forward and backward search – on EL for use in DL were considered and a comprehensive literature search was conducted. First, the basic terms in the domain of EL were introduced and then the relevant combination strategies were described. Subsequently, the individual voting and averaging procedures were added, and an example of application within a CNN was given. Finally, the strengths and limitations of EL were discussed. The structure of this paper results from the fact that a basic overview of the individual specifics of EL procedures is of elementary importance to act purposefully in the application. In addition, research gaps for future research in this domain were identified. Future work should systematically address the formulated research gaps.

References

[1] Y. Chen, Y. Wang, Y. Gu, X. He, P. Ghamisi, and X. Jia, "Deep Learning Ensemble for Hyperspectral Image Classification," *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, vol. 12, no. 6, pp. 1882–1897, Jun. 2019, doi: 10.1109/JSTARS.2019.2915259.

[2] H. Greenspan, B. Van Ginneken, and R. M. Summers, "Guest Editorial Deep Learning in Medical

Imaging: Overview and Future Promise of an Exciting New Technique," *IEEE Trans. Med. Imaging*, vol. 35, no. 5, pp. 1153–1159, May 2016, doi: 10.1109/TMI.2016.2553401.

[3] W. Liu, M. Zhang, Z. Luo, and Y. Cai, "An Ensemble Deep Learning Method for Vehicle Type Classification on Visual Traffic Surveillance Sensors," *IEEE Access*, vol. 5, pp. 24417–24425, Oct. 2017, doi: 10.1109/ACCESS.2017.2766203.

[4] I. Kononenko, "Machine learning for medical diagnosis: history, state of the art and perspective," *Artif. Intell. Med.*, vol. 23, no. 1, pp. 89–109, Aug. 2001, doi: 10.1016/S0933-3657(01)00077-X.

[5] J. Latif, C. Xiao, A. Imran, and S. Tu, "Medical imaging using machine learning and deep learning algorithms: A review," *2019 2nd Int. Conf. Comput. Math. Eng. Technol. iCoMET 2019*, Mar. 2019, doi: 10.1109/ICOMET.2019.8673502.

[6] J. Bell, "What is machine learning?," *Mach. Learn. City Appl. Archit. Urban Des.*, pp. 209–216, May 2022, doi: 10.1007/978-3-319-18305-3_1.

[7] C. Bin Ha and H. K. Song, "Signal Detection Scheme Based on Adaptive Ensemble Deep Learning Model," *IEEE Access*, vol. 6, pp. 21342–21349, Apr. 2018, doi: 10.1109/ACCESS.2018.2825463.

[8] Y. Ren, L. Zhang, and P. N. Suganthan, "Ensemble Classification and Regression-Recent Developments, Applications and Future Directions [Review Article]," *IEEE Comput. Intell. Mag.*, vol. 11, no. 1, pp. 41–53, Feb. 2016, doi: 10.1109/MCI.2015.2471235.

[9] S. J. Lee, T. Chen, L. Yu, and C. H. Lai, "Image Classification Based on the Boost Convolutional Neural Network," *IEEE Access*, vol. 6, pp. 12755–12768, Jan. 2018, doi: 10.1109/ACCESS.2018.2796722.

[10] Z.-H. Zhou, "Ensemble Learning," *Encycl. Biometrics*, pp. 270–273, 2009, doi: 10.1007/978-0-387-73003-5_293.

[11] L. Wen, L. Gao, and X. Li, "A New Snapshot Ensemble Convolutional Neural Network for Fault Diagnosis," *IEEE Access*, vol. 7, pp. 32037–32047, 2019, doi: 10.1109/ACCESS.2019.2903295.

[12] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," *Nat. 2015 5217553*, vol. 521, no. 7553, pp. 436–444, May 2015, doi: 10.1038/nature14539.

[13] X. Dong, Z. Yu, W. Cao, Y. Shi, and Q. Ma, "A survey on ensemble learning," *Front. Comput. Sci.*, vol. 14, no. 2, pp. 241–258, Apr. 2020, doi: 10.1007/S11704-019-8208-Z.

[14] Y. Wang, Y. Yang, Y. X. Liu, and A. A. Bharath, "A Recursive Ensemble Learning Approach with Noisy Labels or Unlabeled Data," *IEEE Access*, vol. 7, pp. 36459–36470, 2019, doi: 10.1109/ACCESS.2019.2904403.

[15] F. Huang, J. Lu, J. Tao, L. L. Li, X. Tan, and P. Liu, "Research on Optimization Methods of ELM Classification Algorithm for Hyperspectral Remote Sensing Images," *IEEE Access*, vol. 7, pp. 108070–108099, 2019, doi: 10.1109/ACCESS.2019.2932909.

[16] O. Sagi and L. Rokach, "Ensemble learning: A survey," *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.*, vol. 8, no. 4, p. e1249, Jul. 2018, doi: 10.1002/WIDM.1249.

[17] "Machine Learning Research: Four Current Directions | Request PDF." Available at : researchgate.net.

[18] S. Wan and H. Yang, "Comparison among methods of ensemble learning," *Proc. - 2013 Int. Symp. Biometrics Secur. Technol. ISBAST 2013*, pp. 286–290, 2013, doi: 10.1109/ISBAST.2013.50.

[19] F. Huang, G. Xie, and R. Xiao, "Research on ensemble learning," *2009 Int. Conf. Artif. Intell. Comput. Intell. AICI 2009*, vol. 3, pp. 249–252, 2009, doi: 10.1109/AICI.2009.235.

[20] L. Breiman, "Bagging predictors," *Mach. Learn.*, vol. 24, no. 2, pp. 123–140, 1996, doi: 10.1007/BF00058655.

[21] Y. H. Na, H. Jo, and J. B. Song, "Learning to grasp objects based on ensemble learning combining simulation data and real data," *Int. Conf. Control. Autom. Syst.*, vol. 2017-October, pp. 1030–1034, Dec. 2017, doi: 10.23919/ICCAS.2017.8204368.

[22] M. A. Dede, E. Aptoula, and Y. Genc, "Deep Network Ensembles for Aerial Scene Classification,"

IEEE Geosci. Remote Sens. Lett., vol. 16, no. 5, pp. 732–735, May 2019, doi: 10.1109/LGRS.2018.2880136.

[23] G. I. Webb and Z. Zheng, "Multistrategy ensemble learning: Reducing error by combining ensemble learning techniques," *IEEE Trans. Knowl. Data Eng.*, vol. 16, no. 8, pp. 980–991, Aug. 2004, doi: 10.1109/TKDE.2004.29.

[24] H. Guan and X. Xue, "Robust online visual tracking via a temporal ensemble framework," *Proc. - IEEE Int. Conf. Multimed. Expo*, vol. 2016-August, Aug. 2016, doi: 10.1109/ICME.2016.7552969.

[25] N. Yu, L. Qian, Y. Huang, and Y. Wu, "Ensemble Learning for Facial Age Estimation Within Non-Ideal Facial Imagery," *IEEE Access*, vol. 7, pp. 97938–97948, 2019, doi: 10.1109/ACCESS.2019.2928843.

[26] J. Zilly, J. M. Buhmann, and D. Mahapatra, "Glaucoma detection using entropy sampling and ensemble learning for automatic optic cup and disc segmentation," *Comput. Med. Imaging Graph.*, vol. 55, pp. 28–41, Jan. 2017, doi: 10.1016/J.COMPMEDIMAG.2016.07.012.

[27] S. A. Gyamerah, P. Ngare, and D. Ikpe, "On Stock Market Movement Prediction Via Stacking Ensemble Learning Method," *CIFEr 2019 - IEEE Conf. Comput. Intell. Financ. Eng. Econ.*, May 2019, doi: 10.1109/CIFER.2019.8759062.

[28] S. Džeroski and B. Ženko, "Is combining classifiers with stacking better than selecting the best one?," *Mach. Learn.*, vol. 54, no. 3, pp. 255–273, Mar. 2004, doi: 10.1023/B:MACH.0000015881.36452.6E.

[29] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, Oct. 2001, doi: 10.1023/A:1010933404324`.

[30] R. Buettner, S. Sauer, C. Maier, and A. Eckhardt, "Towards ex ante prediction of user performance: A novel NeuroIS methodology based on real-time measurement of mental effort," *Proc. Annu. Hawaii Int. Conf. Syst. Sci.*, vol. 2015-March, pp. 533–542, Mar. 2015, doi: 10.1109/HICSS.2015.70.

[31] H. Liang, L. Song, and X. Li, "The rotate stress of steam turbine prediction method based on stacking ensemble learning," *Proc. IEEE Int. Symp. High Assur. Syst. Eng.*, vol. 2019-January, pp. 146–149, Mar. 2019, doi: 10.1109/HASE.2019.00030.

[32] A. P. Piotrowski and J. J. Napiorkowski, "A comparison of methods to avoid overfitting in neural networks training in the case of catchment runoff modelling," *J. Hydrol.*, vol. 476, pp. 97–111, Jan. 2013, doi: 10.1016/J.JHYDROL.2012.10.019.

[33] T. G. Dietterich, "Ensemble methods in machine learning," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 1857 LNCS, pp. 1–15, 2000, doi: 10.1007/3-540-45014-9_1.

[34] Y. Freund and R. E. Schapire, "A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting," *J. Comput. Syst. Sci.*, vol. 55, no. 1, pp. 119–139, Aug. 1997, doi: 10.1006/JCSS.1997.1504.

[35] B. Zhang, Y. Yang, C. Chen, L. Yang, J. Han, and L. Shao, "Action Recognition Using 3D Histograms of Texture and A Multi-Class Boosting Classifier," *IEEE Trans. Image Process.*, vol. 26, no. 10, pp. 4648–4660, Oct. 2017, doi: 10.1109/TIP.2017.2718189.

[36] Z. H. Zhou, J. Wu, and W. Tang, "Ensembling neural networks: Many could be better than all," *Artif. Intell.*, vol. 137, no. 1–2, pp. 239–263, May 2002, doi: 10.1016/S0004-3702(02)00190-X.

[37] Y. Zhao, J. Li, and L. Yu, "A deep learning ensemble approach for crude oil price forecasting," *Energy Econ.*, vol. 66, pp. 9–16, Aug. 2017, doi: 10.1016/J.ENECO.2017.05.023.

[38] G. Huang, Y. Li, G. Pleiss, Z. Liu, J. E. Hopcroft, and K. Q. Weinberger, "Snapshot Ensembles: Train 1, get M for free," *5th Int. Conf. Learn. Represent. ICLR 2017 - Conf. Track Proc.*, Apr. 2017, Accessed: Apr. 18, 2023. [Online]. Available at: https://arxiv.org/abs/1704.00109v1

[39] Y. Freund, "Boosting a Weak Learning Algorithm by Majority," *Inf. Comput.*, vol. 121, no. 2, pp. 256–285, Sep. 1995, doi: 10.1006/INCO.1995.1136.

[40] E. Bauer and R. Kohavi, "Empirical comparison of voting classification algorithms: bagging,

boosting, and variants," *Mach. Learn.*, vol. 36, no. 1, pp. 105–139, 1999, doi: 10.1023/A:1007515423169.

[41] J. R. Quinlan, "Bagging, Boosting, and C4.5," 1996. Available at : semanticscholar.org.

[42] B. Efron and R. J. Tibshirani, "An Introduction to the Bootstrap," *An Introd. to Bootstrap*, May 1994, doi: 10.1201/9780429246593.

[43] A. Kabir, C. Ruiz, and S. A. Alvarez, "Mixed Bagging: A Novel Ensemble Learning Framework for Supervised Classification Based on Instance Hardness," *Proc. - IEEE Int. Conf. Data Mining, ICDM*, vol. 2018-November, pp. 1073–1078, Dec. 2018, doi: 10.1109/ICDM.2018.00137.

[44] B. Krawczyk and M. Woźniak, "Wagging for Combining Weighted One-class Support Vector Machines," *Procedia Comput. Sci.*, vol. 51, no. 1, pp. 1565–1573, Jan. 2015, doi: 10.1016/J.PROCS.2015.05.351.

[45] G. I. Webb, "MultiBoosting: a technique for combining boosting and wagging," *Mach. Learn.*, vol. 40, no. 2, pp. 159–196, Aug. 2000, doi: 10.1023/A:1007659514849.

[46] D. H. Wolpert, "Stacked generalization," *Neural Networks*, vol. 5, no. 2, pp. 241–259, Jan. 1992, doi: 10.1016/S0893-6080(05)80023-1.

[47] T. K. Ho, "The random subspace method for constructing decision forests," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 8, pp. 832–844, 1998, doi: 10.1109/34.709601.

[48] Y. Lin and Y. Jeon, "Random Forests and Adaptive Nearest Neighbors,", vol. 101, no. 474, pp. 578–590, Jun. 2012, doi: 10.1198/016214505000001230.

[49] I. Loshchilov and F. Hutter, "SGDR: Stochastic Gradient Descent with Warm Restarts," *5th Int. Conf. Learn. Represent. ICLR 2017 - Conf. Track Proc.*, Aug. 2016, Accessed: Apr. 18, 2023. [Online]. Available: https://arxiv.org/abs/1608.03983v5

[50] P. K. Chan and S. J. Stolfo, "A Comparative Evaluation of Voting and Meta-learning on Partitioned Data," *Mach. Learn. Proc. 1995*, pp. 90–98, Jan. 1995, doi: 10.1016/B978-1-55860-377-6.50020-7.

[51] J. Zheng, X. Cao, B. Zhang, X. Zhen, and X. Su, "Deep Ensemble Machine for Video Classification," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 30, no. 2, pp. 553–565, Feb. 2019, doi: 10.1109/TNNLS.2018.2844464.