Stochastic and deterministic optimization methods and their applications

Egyetemi doktori (PhD) értekezés

TIBA ATTILA
TÉMAVEZETŐ: DR. HAJDU ANDRÁS

DEBRECENI EGYETEM
Természettudományi és Informatikai Doktori Tanács
Informatikai Tudományok Doktori Iskola
Debrecen, 2022

Ezen értekezést a Debreceni Egyetem Természettudományi és Informatikai Doktori Tanács Informatikai Tudományok Doktori Iskola Diszkrét matematika, adatfeldolgozás és vizualizáció programja keretében készítettem a Debreceni Egyetem műszaki doktori (PhD) fokozatának elnyerése céljából. Nyilatkozom arról, hogy a tézisekben leírt eredmények nem képezik más PhD disszertáció részét, értekezést korábban más intézményben nem nyújtottam be, és azt nem utasították el.

Debrecen, 2022. .....

a jelölt aláírása

Tanúsítom, hogy Tiba Attila doktorjelölt 2016 - 2020 között a fent megnevezett Doktori Iskola Diszkrét matematika, adatfeldolgozás és vizualizáció programjának keretében irányításommal végezte munkáját. Az értekezésben foglalt eredményekhez a jelölt önálló alkotó tevékenységével meghatározóan hozzájárult. Nyilatkozom továbbá arról, hogy a tézisekben leírt eredmények nem képezik más PhD disszertáció részét, értekezést korábban más intézményben nem nyújtotta be, és azt nem utasították el. Az értekezés elfogadását javasolom.

Debrecen, 2022. .....

a témavezető aláírása

Stochastic and deterministic optimization methods and their applications

Értekezés a doktori (Ph.D.) fokozat megszerzése érdekében az informatikai tudományágban

Írta: Tiba Attila okleveles matematikus és gazdaságinformatikus

Készült a Debreceni Egyetem Informatikai Tudományok doktori iskolája, Diszkrét matematika, adatfeldolgozás és vizualizáció programja keretében

Témavezető: Dr. Hajdu András

Az értekezés bírálói:

Dr. ....

Dr. ....

Dr. ....

A bírálóbizottság:

elnök: Dr. ....

tagok: Dr. ....

Dr. ....

Dr. ....

Dr. ....

Az értekezés védésének időpontja:

Contents

Chapter 1

Introduction

Ensemble-based approaches are very effective in various fields in raising the accuracy of its individual members, when some voting rule is applied for aggregating the individual decisions.

In the first part of this dissertation, we investigate how to find and characterize the ensembles having the highest accuracy if the total cost of the ensemble members is bounded. This question leads to a Knapsack-problem with non-linear and non-separable objective function in binary and multiclass classification scenarios where majority voting is used for aggregation. As the conventional solving methods cannot be applied for this task, a novel stochastic approach was introduced in the binary case where the energy function is discussed as the joint probability function of the member accuracies. We show some theoretical results with respect to the expected ensemble accuracy and its variance also for the multiclass classification problem which can help us to solve the Knapsack-problem.

In the next part of the dissertation, we present some cases of the application of ensemble methods in the field of medicine. First, we show our research on the COVID-19 pandemic, which aimed to predict the COVID-19 epidemic curves (new cases per day) using official epidemiological data utilizing a neural network architecture consisting of interconnected

subnetworks, and then compare and validate the predicted models with the observed data.

Furthermore, we investigate the effectiveness of ensemble methods for convolutional neural network (CNN) models. We present an ensemble-based outlier detection method consisting of CNNs combined with a support vector machine (SVM) classifier. Experiments showed that it makes a majority voting-based decision very accurate in outlier filtering. We evaluate the performance of the proposed method for filtering databases of retinal and skin lesion images.

In the final part of the dissertation, we present deterministic optimization methods for measuring image pattern regularity. Texture analysis has received strong attention from researchers for many years. It is important in several cases to determine whether the texture is regular, or to determine at least the presence of some kind of regularity of the pattern within the texture. For example, examination of the regularity of the pigment network has an important role in recognizing typical/atypical lesion behavior. Co-occurrence matrices as sources of second-order statistical descriptors are commonly used in texture classification tasks. To generate such a matrix, we need a position vector to check possible intensity frequencies in its endpoints. In the dissertation, we propose an efficient algorithm to locate such position vectors according to which the pattern of the texture repeats and thus, the descriptors (Haralick features) derived from the co-occurrence matrix are capable to characterize the regularity of the pattern. Our results show that the proposed approach is capable to suggest position vectors for an efficient co-occurrence matrix based texture analysis. Finally, we provide an algorithm to support the decision on whether some repeatedly occurring pattern in a digital image can be considered to have periodical nature or not.

Chapter 2

A stochastic approach for ensemble pruning under resource constrains

2.1 Introduction

Ensemble-based systems are rather popular in several application fields and are employed to increase the decision accuracy of individual approaches. We also encounter such approaches for pattern recognition purposes [53], using models based on, e.g., neural networks [13] decision trees [47] or other principles [38, 40, 88]. In the most recent results, we can recognize this approach in the design of state-of-the-art convolutional neural networks (such as GoogLeNet, incorporating the Inception module [76]) or the direct combination of them [33]. In the literature, ensemble-based approaches are routinely used to aggregate the outputs of pattern classifiers [5] or detector algorithms [4], usually by some majority voting-based rule. During these efforts, we have also faced perhaps the most negative property of creating ensembles, that is, the increasing demand on resources.

This type of cost may occur as the execution/training time and the working hours needed to create the ensemble components, etc., according to the characteristics of the given problem. Thus, in addition to the primary aim of the composition of the most accurate ensemble, a natural constraint emerges as a cost limitation for that endeavor.

The current literature mostly refers to the selection of an efficient ensemble from a pool of possible members as ensemble pruning [89]. Even if no resource constraints are applied, a subset of possible ensemble members may lead to better performance than selecting all the members. Moreover, the best strategy is to compose an ensemble having such good performing members which also have diverse behavior. To realize this approach, ensemble pruning methods can be categorized in the following three main groups: ordering-, clustering-, and optimization-based pruning. Ordering-based pruning ranks the individual members according to some criterion, and the most highly ranked ones are put into the final ensemble. Clustering-based pruning aims to identify representative prototype individual members to compose the ensemble, while the optimization-based approach sets up an objective function and forms a subset of members by minimizing or maximizing it. There are efforts to complement the basic ensemble pruning models to consider possible resource constraints like training/test execution times or memory/storage space [10, 37] as well. To reach this aim a popular approach is to apply multi-objective evolutionary algorithms, like NSGA-II [15]. NSGA-II is an elitist algorithm that provides fast nondominated sorting and considers density estimation and crowded-comparison to maintain diversity. These positive properties can be exploited in ensemble pruning like in [60]. As the best fits to our single-objective setup we have implemented a general purpose genetic algorithm from [22] (chapters 2-3), and a boosting-based pruning one from [59]. In our comparative analyses we will refer to them as Genetic and Pruning, respectively.

In this work, we analyze a single-constraint task on the resources to compose the most accurate ensemble regarding the energy formed by majority voting as the aggregation rule like in [35]. The constraint we consider corresponds to the training time; however, any other type of resources could be considered. We introduce a novel, theoretically well-founded stochastic approach that considers the energy as the joint probability function of the member accuracies. As our main contribution, we show that this type of knowledge can be efficiently incorporated in any stochastic search process as a stopping rule, since we have the information on the expected accuracy or, alternatively, the probability of finding more accurate ensembles. Our empirical analyses also show that including the stochastic estimation as a stopping rule saves a large amount of search time to build accurate ensembles.

We formulate the resource constraint as a knapsack problem, which provides the opportunity of a precise constraint prescription instead of a simple good price/value expectation considered e.g. in [10, 35, 37] to have small, but relatively accurate ensembles. Basically, we follow an ordering-based approach combined with stochastic sampling to compose the ensembles; however, additionally as a novel contribution we suggest a new heuristics for that. Namely, besides its individual accuracy and cost, we calculate such a usefulness value for each possible member during the selection process that reflects its direct behavior according to the objective function, which is based on the majority voting rule in our case. As we will see, our novel stochastic search method is proven to be very competitive with simulated annealing (SA) [17], and the pruning methods [22, 59]. Also, the proposed heuristic can be successfully inserted into these general stochastic search strategies.

The results presented in this chapter are published in my following publications: [29], [80].

2.2 Basic concepts and notation

Let us consider a pool $\mathcal{D} = \{\mathcal{D}_1, \ldots, \mathcal{D}_n\}$ containing possible ensemble members, where each member $\mathcal{D}_i$ $(i = 1, \ldots, n)$ is characterized by a pair (p_i, t_i) describing its individual accuracy $p_i \in [0, 1]$ and cost $t_i \in \mathbb{R}_{>0}$. The individual accuracies are supposed to be known, e.g., by determining them on some test data and by an appropriate performance metric. In this work, we will focus on the majority voting-based aggregation principle, where the possible ensemble members $\mathcal{D}_i$ $(i = 1, \ldots, n)$ are classifiers (see [50]). In [25], the classis case was discussed where the individual classifiers make true/false (binary) decisions. In this model, a classifier D_i with accuracy p_i is considered as a Bernoulli distributed random variable η_i, that is, $P(\eta_i = 1) = p_i$, $P(\eta_i = 0) = 1 - p_i$ $(i = 1, \ldots, n)$, where $\eta_i = 1$ means the correct classification by D_i. In this case, we obtain that the accuracy of an ensemble $\mathcal{D}' = \{\mathcal{D}_{i_1}, \ldots, \mathcal{D}_{i_\ell}\} \subseteq \mathcal{D}$ of $|\mathcal{D}'| = \ell$ members can be calculated as

$$q_\ell(\mathcal{L}) = \sum_{k=\left\lfloor \frac{\ell}{2} \right\rfloor + 1}^{\ell} \left(\sum_{\substack{\mathcal{I} \subseteq \mathcal{L} \\ |\mathcal{I}| = k}} \prod_{i \in \mathcal{I}} p_i \prod_{j \in \mathcal{L} \setminus \mathcal{I}} (1 - p_j) \right), \tag{2.1}$$

where $\mathcal{L} = \{i_1, \ldots, i_\ell\} \subseteq \mathcal{N} = \{1, \ldots, n\}$ is the index set of $\mathcal{D}'$. As an important practical issue, notice that (2.1) is valid only for independent members to calculate the ensemble accuracy. The dependency of the members can be discovered further by, e.g., using different kinds of diversity measures [26].

Regarding ensemble-based systems, the standard task is to devise the most accurate ensemble from $\mathcal{D}$ for the given energy function. In this thesis, we add a natural constraint of a bounded total cost to this optimization

problem. That is, we have to maximize (2.1) under the cost constraint

$$\sum_{i \in \mathcal{L}} t_i \leq T, \tag{2.2}$$

where the total allowed cost $T \in \mathbb{R}_{>0}$ is a predefined constant. Consequently, we must focus on those subsets $\mathcal{L} \subseteq \mathcal{N}$ with cardinalities $|\mathcal{L}| = \ell \in \{1, \ldots, n\}$ for which (2.2) is fulfilled. Let $\mathcal{L}_0$ denote that index set of cardinality $|\mathcal{L}_0| = \ell_0$, where the global maximum ensemble accuracy is reached. The following lemma states that one can reach $\mathcal{L}_0$ calculating $q_\ell(\mathcal{L})$ for odd values of ℓ only, which results in more efficient computation, since not all the possible subsets should be checked.

Lemma 1. *Let*

$$\max\left(q_\ell(\mathcal{L}) \mid \mathcal{L} \subseteq \mathcal{N}\right) = q_{\ell_0}(\mathcal{L}_0). \tag{2.3}$$

If there exists only one index ℓ_0 for which q_ℓ takes its maximum then ℓ_0 is odd.
If there are several indices satisfying the equation (2.3) then the smallest such index ℓ_0 is odd.

Proof. See Appendix 4.3 for the proof. □

The optimization task defined by the energy function (2.1) and the constraint (2.2) can be interpreted as a typical knapsack problem [58]. Such problems are known to be NP-hard; however, if the energy function is linear and/or separable for the p_i-s, then a very efficient algorithmic solution can be given based on dynamic programming. However, if the energy lacks these properties, the currently available theoretical foundation is rather poor. As some specific examples, we were able to locate investigations of an exponential-type energy function [45], and a remarkably restricted family of nonlinear and nonseparable ones [69]. In [45], an approach based on calculus was made by representing the energy function

by its Taylor series. Unfortunately, it has been revealed that dynamic programming can be applied efficiently only to at most the quadratic member of the series; thus, the remaining higher-order members had to be omitted. This compulsion suggests a large error term if this technique is attempted to be considered generally. Thus, to the best of our knowledge, there is a lack of theoretical instructions/general recommendations to solve knapsack problems in the case of complex energy functions. As our energy (2.1) is also nonlinear and nonseparable, we were highly motivated to develop a well-founded framework for efficient ensemble creation.

As our main contribution, in this thesis, we propose a novel stochastic technique to solve knapsack problems with complex energy functions. Though the model is worked out in detail for (2.1) settled on the majority voting rule, it can be applied also to other energy functions. Our approach is based on the stochastic properties of the energy q_ℓ in (2.1), providing that we have some preliminary knowledge on which distribution its parameters p_i $(i = 1, \ldots, n)$ are coming from. We put a special focus on *beta* distributions that fit practical problems very well. In other words, we estimate the distribution of q_ℓ in terms of its mean and variance. This information can be efficiently incorporated as a stopping rule in stochastic search algorithms, as we demonstrate it e.g. for SA. The main idea here is to be able to stop building ensembles when we can expect that better ones can be found by low probability only.

As a common empirical approach to find the optimal ensemble, the usefulness p_i/t_i $(i = 1, \ldots, n)$ of the possible members are calculated. Then, as deterministic greedy methods, the ensemble is composed of forward/backward selection strategies (see, e.g., [51]). Since the deterministic methods are less efficient – e.g., the greedy one is proven to have 50% accuracy for the simplest knapsack energy $\sum_{i=1}^{n} p_i$ – popular stochastic search algorithms are considered instead, such as SA. As a further contribution, we introduce a novel stochastic search strategy, where the use-

fulness of the components is defined in a slightly more complex way to better fit the investigated energy; the stopping rule can be successfully applied in this approach as well. For the sake of completeness, we will start our theoretical investigation regarding the accuracy of the existing deterministic methods when a cost limitation is also applied.

2.3 Deterministic selection strategies

In this section, we address deterministic selection strategies to build an ensemble that has maximal system accuracy $q_{\ell_0}(\mathcal{L}_0)$, applying the cost limitation. However, since we have 2^n different subsets of elements of a pool of cardinality n, this selection task is known to be NP-hard. To overcome this issue, several selection approaches have been proposed. The common point of these strategies is that in general, they do not assume any knowledge on the proper determination of the classification performance $q_\ell(\mathcal{L})$; rather, they require only the ability to evaluate it. Moreover, to the best of our knowledge, strategies that consider the capability of individual feature accuracies to be modeled by drawing them from a probability distribution, as in our approach, have not yet been proposed.

Based on the above discussion, it seems to be natural to ascertain how the widely applied selection strategies work in our setup. The main difference in our case, in contrast to the general recommendations, is that now we can properly formulate the performance evaluation using the exact functional knowledge of q_ℓ. That is, we can characterize the behavior of the strategy with a strict analysis instead of the empirical tests generally applied.

We start our investigation with greedy selection approaches by discussing them via the forward selection strategy. Here, the most accurate item is selected and put in a subset S first. Then, from the remaining $n-1$ items, the component that maximizes the classification accuracy of

the extended ensemble is moved to S. This procedure is then iteratively repeated; however, if the performance cannot be increased by adding a new component, then S is not extended and the selection stops. The first issue we address is to determine the largest possible error this strategy can lead to in our scenario.

Proposition 2.3.1. *The simple greedy forward selection strategy to build an ensemble that applies the majority voting-based rule has a maximum error rate* $1/2$.

Proof. For the proof, see Appendix 4.4. □

As seen from the proof, the error rate of $1/2$ holds for the forward strategy independent of the time constraint. As a quantitative example, let $p_1 = 0.510$ and $p_2 = p_3 = p_4 = p_5 = 0.505$. With this setup, where $\mathcal{I}_k = \{1, \ldots, k\}$, we have $q_1(\mathcal{I}_1) = p_1 = 0.5100$, $q_3(\mathcal{I}_3) = 0.5100$, and $q_5(\mathcal{I}_5) = 0.5112$, which shows that the greedy forward selection strategy is stuck at the single element ensemble, though a more accurate larger one could be found.

In addition to forward selection, its inverse variant, the backward selection strategy, is also popular. It puts all the components into an ensemble first, and in every selection step, leaves the worst one out to gain maximum ensemble accuracy. As a major difference from the forward strategy, backward selection is efficient in our case if the time constraint is irrelevant. Namely, either the removal of the worst items will lead to an increase in q_ℓ defined in (2.1), or the selection can be stopped without the risk of missing a more accurate ensemble. However, if the time constraint applies, the same maximum error rate can be proved.

Proposition 2.3.2. *The simple greedy backward selection strategy considering the individual accuracy values to build an ensemble that applies the majority voting-based rule has a maximum error rate of 1/2.*

Proof. Proposition 2.3.2 is proved in Appendix 4.5. □

Propositions 2.3.1 and 2.3.2 have shown the worst-case scenarios for the forward and backward selection strategies. However, the greedy approach was applied only regarding the accuracy values of the members, and their execution times were omitted. To consider both the accuracies and execution times of the algorithms in the ensemble pool $\mathcal{D} = \{D_1 = (p_1, t_1), D_2 = (p_2, t_2), \ldots, D_n = (p_n, t_n)\}$, we consider their usefulness in the selection strategies, defined as

$$u_i = p_i/t_i, \quad i = 1, \ldots, n, \tag{2.4}$$

which is a generally used definition to show the price-to-value ratio of an object. The composition of ensembles based on similar usefulness measures has also been efficient, e.g., in sensor networks [51].

After the introduction of the usefulness (2.4), the first natural question to clarify is to investigate whether the validity of the error rate of the deterministic greedy forward and backward selection strategies operating with the usefulness measure holds. Through the following two statements, we will see that the $1/2$ error rates remain valid for both greedy selection approaches.

Corollary 1. *Proposition 2.3.1 remains valid when the forward feature selection strategy operates on the usefulness. Namely, as a worst-case scenario, let* $t_1 = t_2 = \cdots = t_n = T/n$ *be the execution times in the example of the proof of Proposition 2.3.1 while keeping the same* $p_1, p_2, \ldots, p_n$ *values. Then, the selection strategy operates completely in the same way on the* $u_i = p_i/t_i$ *values* $(i = 1, \ldots, n)$ *as on the* p_i *ones, since the* t_i *values are equal. That is, the error rate is* $1/2$ *in the same way.*

Proposition 2.3.3. *The simple greedy backward selection strategy considering the individual usefulness (2.4) to build an ensemble that applies the majority voting-based rule has a maximum error rate of 1/2.*

Proof. The proof is provided in Appendix 4.6. □

We note the analogy between forward and backward selection strategies and approximation algorithms, which find approximate solutions to optimization problems, in particular NP-hard ones. With this respect the most common expectation is to have provable guarantees on the distance of the returned solution to the optimal one [85]. With Propositions 2.3.1, 2.3.2, 2.3.3 and Corollary 1 we exactly address this expectation by giving the worst-case scenarios for these selection strategies. A wider theoretical characterization (regarding e.g. the expected error) would need exhaustive knowledge about the specific member accuracies and cost values. However, in the later chapters we will present many empirical results for these greedy approaches in several scenarios. These results also suggest that a deeper error analysis is expected to be interesting from mainly a theoretical point of view, because other existing and the proposed approaches show remarkably better performance.

The main problem with the above deterministic procedures is that they leave no opportunity to find better performing ensembles. Thus, we move on now to the more dynamic stochastic strategies. Keep in mind that since in our model the distribution of q will be estimated, in any of the selection strategies we can exploit this knowledge as a stopping rule. Namely, even for the deterministic approaches, we can check whether the accuracy of the extracted ensemble is already attractive or whether we should continue and search for a better one.

2.4 Stochastic search algorithms

As the deterministic selection strategies may have poor performance, we investigate stochastic algorithms to address our optimization problem. Such randomized algorithms, where randomization only affects the order of the internal executions, produce the same result on a given input, which can cause the same problem we have found for the deterministic ones. In case of Monte Carlo (MC) algorithms [78], the result of the simulations might change, but they produce the correct result with a certain probability. The accuracy of the MC approach depends on the number of simulations N; the larger N is, the more accurate the algorithm will be. It is important to know how many simulations are required to achieve the desired accuracy. The error of the estimate of the probability failure is found to be $u_{1-\alpha/2}\sqrt{(1-\mathrm{P}_f)/N\mathrm{P}_f}$, where $u_{1-\alpha/2}$ is the $1-\alpha/2$ quantile of the standard normal distribution, and P_f is the true value of the probability of failure.

Simulated annealing (SA), as a variant of the Metropolis algorithm, is composed of two main stochastic processes: generation and acceptance of solutions. SA is a general-purpose, serial search algorithm, whose solutions are close to the global extremum for an energy function within a polynomial upper bound for the computational time and are independent of the initial conditions.

To compare the MC method with SA for solving a knapsack problem, we applied simulations for that scenario in which the deterministic approaches failed to find the most accurate ensemble, that is, when $D_1 = (1-\beta, T)$, and $D_2 = D_3 = \ldots = D_n = (1/2+\varepsilon, T/n)$ with $0 < \beta < 1/2, 0 < \varepsilon < 1/2$. For this setup, we obtained that the precision of the MC method was only 11%, while SA found the most accurate ensemble in 96% of the simulations. Beyond SA, other pruning methods cited in the introduction are naturally based on stochastic methods.

Now, we introduce a novel search strategy that takes better advantage of our stochastic approach than, e.g., SA. This strategy builds ensembles using a randomized search technique and introduces a concept of usefulness for member selection, which better adapts to the ensemble energy than the classic one (2.4). Namely, in our proposed approach, the selection of the items for the ensemble is based on the efficiency of the members determined in the following way: for the i-th item with accuracy p_i and execution time t_i, the system accuracy $q(p_i, t_i)$ of the ensemble containing the maximal number of i-th items

$$q(p_i, t_i) = \sum_{k=\lfloor T/t_i \rfloor/2+1}^{\lfloor T/t_i \rfloor} \binom{\lfloor T/t_i \rfloor}{k} {p_i}^k (1-p_i)^{\lfloor T/t_i \rfloor - k} \tag{2.5}$$

characterizes the efficiency (usefulness) of the i-th item, instead of (2.4).

A greedy algorithm for an optimization problem always chooses the item that seems to be the most useful at that moment. In our selection method, a discrete random variable depending on the efficiency values of the remaining items is applied in each step to determine the probability of choosing an item from the remaining set to add to the ensemble. Namely, in the k-th selection step, if the items $i_1, \ldots, i_{k-1}$ are already in the ensemble, then the efficiency values $q^{(k-1)}(p_i, t_i)$ of the remaining items are updated to the maximum time of $T_k = T - \sum_{j=1}^{k-1} t_{i_j}$, where $q^{(0)}(p_i, t_i) = q(p_i, t_i)$ and $T_0 = T$.

The i-th item is selected as the next member of the ensemble with the following probability:

$$(\mathrm{P}_{ens})_i^{(k)} = \frac{q^{(k-1)}(p_i, t_i)}{\sum_j q^{(k-1)}(p_j, t_j)}, \tag{2.6}$$

where $i, j \in \mathcal{N} \backslash \{i_1, \ldots, i_{k-1}\}$. This discrete random variable reflects that the more efficient the item is, the more probable it is to be selected for the ensemble in the next step.

As a new contribution, we have incorporated these probabilities based on the newly introduced member efficiencies first to SA characterizing the acceptance probabilities by them. Same considerations apply to any other stochastic search methods. Besides SA and the genetic algorithm [22], these novel stochastic methods (denoted by SA+ and Genetic+) will be compared with our proposed method in the experimental analyses.

If $t_i > T_k$ for all $i \in \mathcal{N} \backslash \{i_1, \ldots, i_{k-1}\}$, then our stochastic process ends for the given search step since there is not enough remaining time for any items. Then, we restart the process to extract another ensemble in the next search step. As a formal description of our proposed stochastic search method SHErLoCk, see Algorithm 1; notice that we evaluate the accuracy of ensembles with odd cardinalities only as in Lemma 1. A very important issue regarding both our approach and other search methods (e.g. SA) is the exact definition of the number of search steps, that is, a meaningful STOP parameter – and also an escaping MAXSTEP one – for Algorithm 1. In the forthcoming sections, we present how the proper derivation of the stopping parameters (STOP and MAXSTEP) can be derived.

2.5 Stochastic estimation of ensemble energy

We need to examine and characterize the behavior of q_ℓ in (2.1) to exploit these results to find and apply the proper stopping criteria in stochastic search methods.

Let $p \in [0, 1]$ be a random variable with mean μ_p and variance σ_p^2, where p_i $(i = 1, 2, \ldots, n)$ are independent and identically distributed according to p, i.e., a sample. Furthermore, let μ_{q_ℓ} and $\sigma_{q_\ell}^2$ denote the mean and variance of the ensemble accuracy q_ℓ, respectively. In this case, it is

Algorithm 1 Proposed Stochastic searcH for EnsembLe Creation (SHErLoCk).

Input: Pool $\mathcal{D} = \{(p_i, t_i)\,,\ i = 1, \ldots, n\}$,
Total allowed time T,
Stochastic stopping value STOP,
Maximum search steps MAXSTEP.

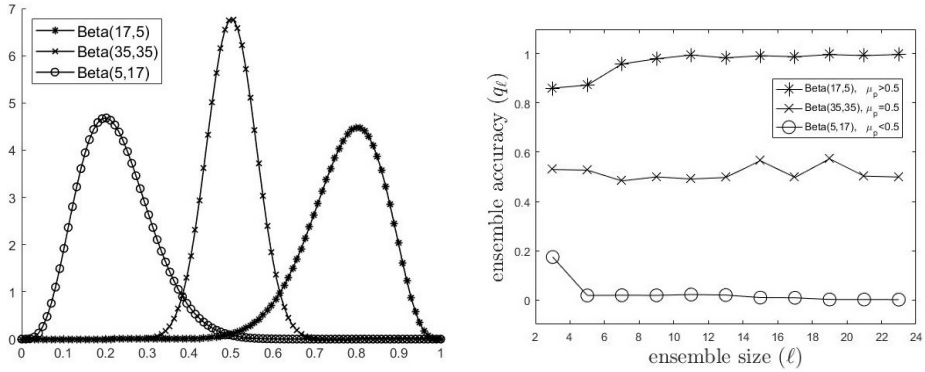
Output: An ensemble MAXENS $\subseteq \mathcal{D}$ to maximize system accuracy (2.1) within time T as in (2.2).

STEP $\leftarrow$ 0, MAXENS $\leftarrow \emptyset$, $q_{\ell_0} \leftarrow 0$
while STEP<MAXSTEP **do**
  $H \leftarrow \mathcal{D}$, ENS $\leftarrow \emptyset$, $T' \leftarrow T$, SP $\leftarrow \emptyset$
  while $\exists (p_j, t_j) \in H : t_j \leq T - \sum_{(p_k,t_k)\in \text{ENS}} t_k$ **do**
    $\forall (p_i, t_i) \in H$ calculate $q(p_i, t_i)$ by (2.5)
    $\forall (p_i, t_i) \in H$ calculate P_{ens_i} by (2.6) and $SP \leftarrow SP \cup \{\text{P}_{ens_i}\}$
    Select a (p_j, t_j) randomly from H by distribution SP
    if $t_j < T'$ **then**
      $ENS \leftarrow ENS \cup \{(p_j, t_j)\}$
      $H \leftarrow H \setminus \{(p_j, t_j)\}$
      $T' \leftarrow T' - t_j$
      if $mod(size(\text{ENS}), 2) = 1$ **then**
        Calculate q_ℓ(ENS) by (2.1)
      end if
      if $q_{\ell_0} < q_\ell$ **then**
        $q_{\ell_0} \leftarrow q_\ell$, MAXENS $\leftarrow$ ENS
      end if
      if $q_{\ell_0} > STOP$ **then**
        return MAXENS
      end if
    end if
  end while
  STEP $\leftarrow$ STEP + 1
end while
return MAXENS

seen that $\mu_p \leq 1$ and a simple calculation shows that

$$\mu_{q_\ell} = \sum_{k=\left\lfloor \frac{\ell}{2} \right\rfloor + 1}^{\ell} \binom{\ell}{k} \mu_p^k \left(1 - \mu_p\right)^{\ell - k}. \tag{2.7}$$

The mean μ_{q_ℓ} is monotonic in ℓ, except for the case $\mu_p = 1/2$. Moreover, we get 0, 1/2 or 1 for the limit of μ_{q_ℓ} if $\ell \to \infty$ in case $\mu_p < 1/2$, $\mu_p = 1/2$, $\mu_p > 1/2$, respectively. As a demonstrative example, see Figure 2.1 regarding the three possible accuracy limits (0, 1/2 or 1) described in (4.18) with respective *Beta*(α_p, β_p) distributions for p. The parameters of the beta distribution (α_p, β_p) can be chosen arbitrarily to fulfil the condition $\mu_p < 1/2$, $\mu_p = 1/2$, $\mu_p > 1/2$, respectively. Furthermore, the variance $\sigma_{q_\ell}^2$ can be expressed by the mean μ_p and variance σ_p^2, where its limit $\lim_{\ell \to \infty} \sigma_{q_\ell}^2 = 0$, if $\mu_p \neq 1/2$ and $s_T = \sigma_p^2 + \mu_p^2 \neq 1/2$ and 1, otherwise. For the exact formula for $\sigma_{q_\ell}^2$ and for the proof of these statements, see Lemma 2 in Appendix 4.7. Notice that the condition $\ell \to \infty$ naturally assumes the same for the pool size with $n \to \infty$.



(a) (b)

Figure 2.1: Different sample *Beta*(α_p, β_p) distributions (a) and the convergence of ensemble accuracies for member accuracies coming from them (b).

Now, to devise a stochastic model, we start with checking the possible distributions of the member accuracy values p_i to estimate the ensemble accuracy. Then, we extend our model regarding this estimation by incorporating time information as well. Notice that the estimation of the ensemble accuracy will be exploited to derive a stopping rule for the ensemble selection process.

2.6 Estimation of the distribution of member accuracies

Among the various possibilities, we have found that the *beta* distribution is a very good choice to analyze the distribution of member accuracies. The main reason is that *beta* concentrates on the interval $[0, 1]$, that is, it can exactly capture the domain for the smallest/largest accuracy. Moreover, the *beta* distribution is able to provide density functions of various shapes that often appear in practice. Thus, to start the formal description, let the variate p be distributed as *Beta*(α_p, β_p) with density

$$b\left(x; \alpha_p, \beta_p\right) = \frac{x^{\alpha_p - 1}\left(1 - x\right)^{\beta_p - 1}}{B\left(\alpha_p, \beta_p\right)}, \tag{2.8}$$

where $B\left(\alpha_p, \beta_p\right) = \Gamma\left(\alpha_p\right)\Gamma\left(\beta_p\right)/\Gamma\left(\alpha_p + \beta_p\right)$. In this case,

$$\mu_p = \alpha_p / \left(\alpha_p + \beta_p\right), \tag{2.9}$$

and $\mu_p \in (1/2, 1)$ if and only if $\alpha_p > \beta_p$. If $\alpha_p = \beta_p$, then $\mu_p = 1/2$. In the case of $\alpha_p > \beta_p$, the mode is also greater than $1/2$. The mode is infinite if $\beta_p < 1$; therefore, we exclude this situation and we assume from now on that

$$1 < \beta_p < \alpha_p. \tag{2.10}$$

The variance of p is

$$\sigma_p^2 = \frac{\alpha_p \beta_p}{\left(\alpha_p + \beta_p\right)^2\left(\alpha_p + \beta_p + 1\right)}. \tag{2.11}$$

Since μ_{q_ℓ}, and $\sigma^2_{q_\ell}$ depend on μ_p, and σ^2_p according to (2.7) and (4.19) respectively, one can calculate both of them explicitly. The convergence of μ_{q_ℓ} to 1 is fast if μ_p is close to 1, i.e., $\beta_p \ll \alpha_p$; for instance, if $\alpha_p = 17$, $\beta_p = 5$. Simulations show that the speed of the convergence of $\sigma^2_{q_\ell}$ is exponential; hence, the usual square-root law does not provide the Central Limit Theorem for q_ℓ.

In practice, we perform a *beta* fit on the p_i's $(i = 1, \ldots, n)$. If a fit is found at least at the confidence level 0.95, we take the parameters α_p, β_p provided by the fit and calculate μ_p, σ^2_p by (2.9) and (2.11), respectively. If the *beta* fit is rejected, then μ_p and σ^2_p are estimated from the p_i's as the empirical mean and variance:

$$\mu_p = \frac{1}{n}\sum_{i=1}^{n} p_i,\ \sigma^2_p = \frac{1}{n-1}\sum_{i=1}^{n}(p_i - \mu_p)^2. \tag{2.12}$$

To simplify our further notation we do not indicate whether the mean and variance have been estimated from the fitted distribution or empirically.

2.6.1 Adding time constraints to the model

Now, we turn to the case when together with the item accuracy p_i, we consider its running time t_i as well. The common distribution of a random time is exponential, so let τ be an exponential distribution with density $\lambda \exp(-\lambda t)$. If p is distributed as *Beta*(α_p, β_p), then with setting $\lambda = 1-p$ for a given p, the distribution of λ becomes *Beta*(β_p, α_p).

This is a reasonable behavior of time because it is quite natural to assume that more accurate components require more resources such as a larger amount of computation times. On the other hand, the selection procedure becomes trivial, if, e.g., the time and accuracy are inversely proportional, since then the most accurate member is also the fastest one; therefore, it should be selected first by following this strategy for the remaining members until reaching the time limit. For some other possible

simple accuracy–time relations, see [30].

For a given time constraint T, consider the random number ℓ_T such that

$$\sum_{j=0}^{\ell_T} \tau_j \leq T. \tag{2.13}$$

We provide an estimation for the expected size of the composed ensemble $\widehat{\ell_T} = \left\lceil T\frac{\beta_p-1}{\alpha_p+\beta_p-1} \right\rceil$ in Lemma 3 (see Appendix 4.8) and we incorporate this information into our stochastic characterization of q_ℓ. Till this point, we have assumed that p is distributed as *beta* to calculate $\widehat{\ell_T}$ by Lemma 3. If this is not the case, we consider the following simple and obvious calculation for the approximate number of ℓ under the time constraint T:

$$\widehat{\ell_T} = \left\lceil nT \Big/ \sum_{i=1}^{n} t_i \right\rceil = \left\lceil T/\bar{t} \right\rceil; \tag{2.14}$$

another alternative to derive $\widehat{\ell_T}$ in this case is discussed in section 2.9. In either way it is derived, the value $\widehat{\ell_T}$ will be used in the stopping rule in our ensemble selection procedure; the proper details will be given next.

2.6.2 Stopping rule for ensemble selection

The procedure of finding $(\mathcal{L}_0, \ell_0)$ is a selection task that is NP-hard. We propose an algorithm such that we stop the selection when the value of $q_\ell(\mathcal{L})$ is sufficiently close to the possible maximum, which is not known. To be able to do so, we must give a proper stochastic characterization of q_ℓ by also settling on the calculation of μ_{q_ℓ} and $\sigma^2_{q_\ell}$ via Lemma 2. First, notice that the values of q_ℓ are in $(0, 1)$; indeed, it is positive and

$$q_\ell = \sum_{k=\left\lfloor \frac{\ell}{2} \right\rfloor+1}^{\ell} \sum_{\substack{I\subseteq\mathcal{N} \\ |I|=k}} \prod_{i\in I} p_i \prod_{j\in\mathcal{N}\setminus I} (1-p_j) < \prod_j (p_j + (1-p_j)) = 1. \tag{2.15}$$

For the case when p_i's are *beta* distributed, the product of independent *beta* variates can be close to *beta* again; see [77]. We have also performed MC simulation and found that *beta* distributions fit q_ℓ particularly well, compared to, e.g., the gamma, normal, Weibull, and extreme-valued distributions. Specifically, though the *beta* behavior of q_ℓ was naturally more stable for *beta* distributed p_i's, the usual behavior of q_ℓ was also the same for non-*beta* p_i's.

Thus, to provide a description of the stochastic behavior of q, we consider the following strategy. With a primary assumption on the *Beta*(α_q, β_q) distribution of q_ℓ, we calculate α_q and β_q as

$$\alpha_q = \left(\frac{1-\mu_q}{\sigma_q^2} - \frac{1}{\mu_q} \right) \mu_q^2, \quad \beta_q = \alpha_q \left(\frac{1}{\mu_q} - 1 \right). \tag{2.16}$$

If time information is provided for the pool items, we calculate $\widehat{\ell_T}$ by Lemma 3, and as a simpler notation, we will write $\widehat{\ell}$ from now on. If time information is not available, we will set $\widehat{\ell} = n$.

Next, we decide whether q_ℓ should be considered as *beta* with requiring $1 < \beta_q < \alpha_q$ to be fulfilled to have a mode that is larger than 1/2 and finite. If this condition does not hold, we reject the *beta* behavior of q_ℓ, and based on simulations, we characterize it as a normal distribution and stop the search if

$$q_\ell \geq \kappa_{0.9} \sigma_{q_{\widehat{\ell}}} / \sqrt{\widehat{\ell}} + \mu_{q_{\widehat{\ell}}} = \text{STOP}, \tag{2.17}$$

where $\kappa_{0.9}$ is the 0.9 quantile of the standard normal distribution. Otherwise, when q_ℓ is considered *beta*, we calculate the mode ν of *Beta*(α_q, β_q) for q_ℓ as

$$\nu = \frac{\alpha_q - 1}{\alpha_q + \beta_q - 2}, \tag{2.18}$$

and the Pearson's first skewness coefficient as

$$\gamma = \frac{1-\nu}{\sigma_{q_{\widehat{\ell}}}}. \tag{2.19}$$

Then, we use Table 2.1 to select the appropriate probability value $\varrho_{q_{\widehat{\ell}}}$; the entries are determined by simulation in the case of $2 \leq \beta_q < \alpha_q$.

We stop the selection when the ensemble accuracy reaches the value of the inverse cumulative distribution $F^{-1}_{\alpha_q,\beta_q}(\varrho_{q_{\widehat{\ell}}})$ of $\mathit{Beta}(\alpha_q, \beta_q)$ in the given probability, that is, when

$$q_\ell \geq F^{-1}_{\alpha_q,\beta_q}(\varrho_{q_{\widehat{\ell}}}) = \text{STOP}. \tag{2.20}$$

Table 2.1: Probability values $\varrho_{q_{\widehat{\ell}}}$ for stopping thresholds for different skewness coefficients γ.

γ	$\varrho_{q_{\widehat{\ell}}}$
$\gamma \leq 1$	0.6
$1 < \gamma \leq 2.5$	0.8
$2.5 < \gamma \leq 3.5$	0.9
$3.5 < \gamma$	0.95

In either via (2.17) or (2.20), an estimation for the ensemble accuracy is gained; we obtain a STOP value to stop the stochastic search. However, there is some chance that STOP is not exceeded, though in our experiments it has never occurred. Thus, to avoid an infinite loop, we consider a maximum allowed step number MAXSTEP as an escaping stopping rule. Namely, to obtain MAXSTEP, we apply Stirling's approximation

$$\text{MAXSTEP} = \binom{n}{\widehat{\ell}} \sim n^{\widehat{\ell}}/\widehat{\ell}!, \tag{2.21}$$

assuming that $\widehat{\ell}/n \to 0$. This is a reasonable approach since $\widehat{\ell}$ is calculated according to Lemma 3 or (2.14). The formal description of our proposed ensemble selection method is enclosed in Algorithm 2.

Algorithm 2 Proposed Ensemble Creation Method.

Input: [NO-TIME]: Pool $\mathcal{D} = \{p_i\}_{i=1}^n$.

Input: [TIME]: Pool $\mathcal{D} = \{(p_i, t_i)\}_{i=1}^n$,
Total allowed time T.

Output: An ensemble MAXENS $\subseteq \mathcal{D}$ to maximize system accuracy (2.1) within time T as in (2.2).

Calculate the mean μ_p and std σ_p for $\{p_i\}_{i=1}^n$ by (2.9) and (2.11) (if a *beta* fits to p) or empirically (if p is not *beta*) by (2.12)
switch Input do
  case NO-TIME
    $\widehat{\ell} \leftarrow n$
  case TIME
    Estimate # of members $\widehat{\ell}$ for T by Lemma 3 if a *beta* fits to p, or by (2.14) if p is not *beta*
Calculate $\mu_{q_{\widehat{\ell}}}$ by (2.7) and $\sigma^2_{q_{\widehat{\ell}}}$ by (4.19)
Calculate α_q, β_q by (2.16)
if $1 < \beta_q < \alpha_q$ **then**
  Calculate cdf. F_{α_q,β_q}, ν, γ, $\varrho_{q_{\widehat{\ell}}}$ by (2.18), (2.19) and Table 2.1, and adjust STOP with (2.20)
else
  Adjust STOP with (2.17)
end if
Calculate MAXSTEP by (2.21)
switch Input **do**
  case NO-TIME
    Compose ensemble by SA/Genetic/Pruning using STOP for the stopping rule
  case TIME
    Compose ensemble either by Algorithm 1 or SA/SA+/Genetic/Genetic+/Pruning using STOP for the stopping rule

Notice that neither Algorithm 1 nor Algorithm 2 considers freely adjustable parameters beyond the input (p_i, t_i) pairs and the total allowed time T. The derivation of all estimated distribution parameters and the stopping related ones are properly referred to in the bodies of the algorithms. If no time condition is provided then any preferred unconstrained algorithm (e.g. the SA, Genetic [22] or Pruning [59] one) can be used as handled by line 17 in Algorithm 2. Similarly, the output (line 19) of Algorithm 2 is the composed ensemble found by either our proposed method or any other preferred one again. Either time condition is provided or not, all the ensemble creator approaches can take advantage of the calculated stopping parameter STOP.

Before providing our detailed empirical results in section 2.7, in Table 2.2 we summarize our findings for Algorithm 2 on simulations. Namely, in two respective demonstrative tests with $i = 1, \ldots, 30$ and $i = 1, \ldots, 100$, we have generated the p_i's to come from the same example $\mathit{Beta}(17, 5)$ as before and the execution times t_i from conditional exponential distributions with parameters $\lambda = 1 - p_i$. The time constraint T was set in seconds to 30% of the total time $\sum_{i=1}^{30} t_i$ for the first, and 20% of $\sum_{i=1}^{100} t_i$ for the second test. Both tests were repeated 100 times, and we have taken the averages of the obtained precisions. The parameters of the beta distribution for the simulations can be chosen arbitrarily to fulfil the condition $1 < \beta_p < \alpha_p$ derived in section 2.6. As our primary aim, we have checked whether the stopping rule of the stochastic search indeed led to a reasonable computational gain. For the sake of completeness, in Table 2.2 we have also shown the results for these simulated pairs (p_i, t_i) regarding letting the search continue in the long run (stopped by MAXSTEP), though in each of our tests, the STOP value has been exceeded much earlier. Secondarily, we have compared SA with our selection method SHErLoCk given in Algorithm 1. For Table 2.2, we can conclude that applying our stopping rule by using STOP saved considerable computational

time compared with the exhaustive search that culminated by stopping it with MAXSTEP with a negligible drop in accuracy. Moreover, our approach has found efficient ensembles quicker than SA. These impressions have also been confirmed by the empirical evaluations on real data described in the next section.

Table 2.2: Result of Algorithm 2 on simulations.

Search method	Ensemble accuracy		Comp. time (secs)	
	MAXSTEP	STOP	MAXSTEP	STOP
SHErLoCk (n=30)	99.56%	99.39%	60.03	0.08
SA (n=30)	98.97%	98.91%	87.40	0.30
SHErLoCk (n=100)	99.66%	99.61%	294.58	1.54
SA (n=100)	99.38%	99.37%	638.39	1.58

2.7 Empirical analysis

In this section, we demonstrate the efficiency of our models through an exhaustive experimental test on publicly available data. Our first experiment considers the possibility of organizing competing approaches with different accuracies into an ensemble. In this scenario, accuracy values correspond to final scores of participants of Kaggle[1] challenges without cost/time information provided. Our second setup for ensemble creation considers machine learning-based binary classifiers as possible members; the performance evaluation is performed on several UCI Machine Learning Repository [16] datasets with the training times considered as costs.

[1] www.kaggle.com

2.7.1 Kaggle challenges

Kaggle is an open online platform for predictive modeling and analytics competitions with the aim of solving real-world machine learning problems provided by companies or users. The main idea behind this crowdsourcing approach is that a countless number of different strategies might exist to solve a specific task, and it is not possible to know beforehand which one is the most effective. Though primarily only the scores of the participating algorithms can be gathered from the Kaggle site, as a possible future direction, we are curious regarding whether creating ensembles from the various strategies could lead to an improvement regarding the desired task.

Not all of the Kaggle competitions are suitable to test our models since in the current content, we focus on majority voting-based ensemble creation. Consequently, we have collected only such competitions and corresponding scores where majority voting-based aggregation could take place. More precisely, we have restricted our focus only to such competition metrics based on which majority voting can be realized. Such metrics include quadratic weighted kappa, area under the ROC curve (AUC), log loss, normalized Gini coefficient. For concrete competitions where these metrics were applied, we analyze the following ones: Diabetic Retinopathy Detection[2], DonorsChoose.org Application Screening [3], Statoil/C-CORE Iceberg Classifier Challenge[4], WSDM - KKBox's Churn Prediction Challenge[5], and Porto Seguro's Safe Driver Prediction[6].

For our analytics, on the one hand it is interesting to observe the distribution of the final score of the competitors, which is often affected by

[2]www.kaggle.com/c/diabetic-retinopathy-detection

[3]www.kaggle.com/c/donorschoose-application-screening

[4]www.kaggle.com/c/statoil-iceberg-classifier-challenge

[5]www.kaggle.com/c/kkbox-churn-prediction-challenge

[6]www.kaggle.com/c/porto-seguro-safe-driver-prediction/data

the volume of the prize money offered to the winner. Moreover, accuracy measurement is usually scaled to the interval $[0, 1]$, with 0 for the worst and 1 for the perfect performance, which allows us to test our results regarding the *beta* distributions. As a drawback of Kaggle data, access to the resource constraints corresponding to the competing algorithms (e.g., training/execution times) is rather limited; such data are provided for only a few competitions, primarily in terms of execution time interval.

Thus, to summarize our experimental setup, we interpret the competing solutions of a Kaggle challenge as the pool $\{D_1, D_2, \ldots, D_n\}$, where the score of D_i is used for the accuracy term $p_i \in [0, 1]$ in our model. Then, we apply a *beta* fit for each investigated challenge to determine whether a *beta* distribution fits the corresponding scores or not. If the test is rejected, we can still use the estimation for the joint behavior q using (2.12) and (2.14). If the *beta* test is accepted, we can also apply our corresponding results using (2.9), (2.11), and Lemma 3. Notice that reliably fitting a model for the scores of the competitors might lead to a better insight of the true behavior of the data of the given field, also for the established expectations there.

As observed from Table 2.3, SA was able to stop much earlier with a slight loss in accuracy using the suggested stopping rule (STOP) in finding the optimal ensemble. Our approach SHErLoCk given in Algorithm 1 has been excluded from this analysis since no cost information was available. Though for the lack of cost information our stochastics-based results can be applied only partially to Kaggle challenges with distribution fitting and suggesting stopping criterion accordingly, we were highly motivated to include this platform as well. Kaggle collects a huge number of different approaches – sometimes also with available implementations – for the same task, so is an excellent platform to create ensembles. Moreover, several accuracy measures considered in these challenges are just completely suitable for stochastic analysis, just like in our approach. If resource in-

formation is also provided in the future for the submitted solutions, then our corresponding results also become applicable.

Table 2.3: Ensemble accuracies on the Kaggle datasets found by simulated annealing (SA).

Dataset	Ensemble accuracy		Computational time (secs)	
Name	MAXSTEP	STOP	MAXSTEP	STOP
Diabetic Retinopathy Detection	94.34%	93.19%	194.12	1.31
DonorsChoose.org Application Screening	94.78%	91.96%	206.89	1.67
Statoil/C-CORE Iceberg Classifier Challenge	88.42%	87.76%	191.91	2.23
WSDM - KKBox's Churn Prediction Challenge	96.96%	96.32%	203.88	1.45
Porto Seguro's Safe Driver Prediction	92.99%	89.98%	214.28	1.95
Average	92.29%	90.45%	202.21	1.72

2.7.2 Binary classification problems

The UCI Machine Learning Repository [16] is a popular platform to test the performances of machine learning-based approaches, primarily for classification purposes. A large number of datasets are made publicly available here among which our models can be tested on binary classification ones. That is, in this experiment, the members $D_1, D_2, \ldots, D_n$ of a pool for ensemble creation are interpreted as binary classifiers, whose outputs can be aggregated by the majority voting rule. Using the ground truth supplied with the datasets, the accuracy term $p_i \in [0, 1]$ stands for the classification accuracy of D_i.

The number of commonly applied classifiers is relatively low; therefore to increase the cardinality of the pool, we have also considered a synthetic approach in a similar way to [11]. Namely, we have trained the same base classifier on different training datasets, by which we can synthesize several "different" classifiers. Naturally, this method is able to provide more independent classifiers only if the base classifier is unstable, i.e., minor changes in the training set can lead to major changes in the

classifier output; such an unstable classifier is, for example, the perceptron one.

To summarize our experimental setup for UCI binary classification problems, we have considered base classifiers perceptron [18], decision tree [66], Levenberg-Marquardt feedforward neural network [73], random neural network [82], and discriminative restricted Boltzmann machine classifier [54] for the UCI datasets MAGIC Gamma Telescope, HIGGS, EEG Eye State, Musk (Version 2), Spambase, Breast Cancer Wisconsin, Mushroom, Gisette and Adult; datasets of large cardinalities were selected to be able to train synthetic variants of base classifiers on different subsets. To check our models for different numbers of possible ensemble members, the respective pool sizes were set to $n = 30$ and $n = 100$; the necessary number of classifiers has been reached via synthesizing the base classifiers with training them on different subsets of the training part of the given datasets. In contrast to the Kaggle challenges, in these experiments we were able to retrieve meaningful cost information to devise a knapsack scenario. Namely, for a classifier D_i, its training time was adjusted as its cost t_i in our model. Notice that for even the same classifier, it was possible to obtain different t_i values with training its synthetic variants on datasets of different sizes. Using this time information, for the estimated size $\widehat{\ell}$ of the optimal ensemble, we could use Lemma 3 for $n = 30$, while (2.14) for the case $n = 100$. This is one of the reasons why we set the different pool sizes to $n = 30$ and $n = 100$. Another point is to get sufficiently large search spaces to show the efficiency of our proposed method. To choose a pool size greater than 100 is rather unrealistic and results in a very time-demanding problem.

We compare the performance of the proposed search strategy (SHErLoCk) with SA, SA+, Genetic [22], Genetic+, Pruning [59] on binary classification problems of UCI datasets using an ensemble pool of $n = 30$ and $n = 100$ classifiers, respectively. As clearly visible from Tables 2.4 and

Table 2.4: Comparing the proposed search strategy (SHErLoCk) with other selection methods on binary classification problems of UCI datasets using an ensemble pool of $n = 30$ classifiers.

		Dataset (size)	MAGIC (19 020)	Spambase (4 601)	HIGGS (20 000)	EEG (14 980)	Musk (6 598)	Breast (699)	Mushroom (8124)	Gisette (13 500)	Adult (48 842)	Average
		Method										
Ensemble accuracy	MAXSTEP	SHErLoCk	**93.87%**	**97.26%**	76.19%	96.30%	99.20%	**98.08%**	**99.90%**	**93.89%**	**86.92%**	**93.52%**
		SA+	93.34%	96.68%	76.23%	**96.61%**	99.25%	97.01%	99.53%	93.02%	86.10%	93.09%
		SA	93.16%	96.68%	**76.26%**	96.09%	98.04%	97.39%	98.58%	93.05%	85.57%	92.76%
		Genetic+	93.86%	97.12%	75.39%	96.07%	98.99%	97.74%	99.43%	93.14%	87.05%	93.20%
		Genetic	**93.87**%	96.68%	76.12%	96.19%	**99.27%**	97.23%	99.53%	93.39%	86.91%	93.24%
		Pruning	92.83%	96.06%	75.51%	95.33%	98.06%	97.58%	98.24%	92.37%	85.37%	92.37%
	STOP	SHErLoCk	92.29%	95.47%	**74.63**%	**95.43**%	**99.02**%	**97.53**%	98.58%	93.03%	**85.67**%	**92.42%**
		SA+	**92.67%**	95.95%	74.09%	95.02%	98.06%	96.75%	98.68%	92.48%	85.03%	92.08%
		SA	92.13%	**95.98%**	74.23%	95.31%	98.01%	96.98%	98.44%	93.01%	85.06%	92.13%
		Genetic+	92.04%	95.31%	74.07%	95.16%	98.84%	96.22%	98.35%	92.96%	85.38%	92.04%
		Genetic	92.04%	95.11%	74.58%	95.11%	98.51%	96.20%	**98.79%**	**93.05%**	**85.67%**	92.12%
		Pruning	92.07%	95.12%	73.85%	95.01%	98.03%	96.57%	98.39%	90.13%	84.84%	91.56%
	DET	Forward	90.41%	93.82%	70.42%	94.96%	95.14%	96.22%	98.08%	91.38%	83.49%	90.43%
		Backward	90.19%	93.82%	68.76%	94.96%	95.14%	95.98%	98.08%	91.38%	82.98%	90.14%
Computational time (secs)	MAXSTEP	SHErLoCk	46.47	**37.90**	**49.46**	**30.20**	**48.89**	**32.19**	**37.06**	**51.45**	**34.73**	**40.92**
		SA+	121.9	100.98	128.90	145.57	108.21	101.73	106.13	132.43	96.51	115.82
		SA	79.06	80.12	93.47	71.39	67.26	77.25	60.19	95.98	72.28	77.44
		Genetic+	63.42	68.03	59.57	61.07	74.02	69.09	89.67	67.03	69.08	69.00
		Genetic	**46.08**	48.92	53.22	58.16	51.23	48.56	57.99	64.42	49.18	53.08
		Pruning	125.84	107.08	115.42	129.45	93.06	90.98	103.54	137.08	151.08	117.06
	STOP	SHErLoCk	**0.98**	**0.39**	**0.93**	**0.38**	**0.92**	**0.74**	**0.39**	**0.4**3	**0.28**	**0.60**
		SA+	8.41	4.67	6.73	8.47	7.33	5.09	8.79	5.88	6.11	6.83
		SA	8.78	6.92	4.56	7.93	8.59	9.48	6.95	5.89	7.99	7.45
		Genetic+	5.53	5.96	6.45	9.97	5.15	7.56	6.93	7.91	5.69	6.79
		Genetic	6.78	6.23	6.49	10.98	8.52	10.43	6.02	5.62	8.43	7.72
		Pruning	13.61	12.21	14.91	18.85	11.07	12.98	14.03	13.57	12.09	13.70
	DET	Forward	0.28	0.19	0.41	0.29	0.28	0.38	0.46	0.49	0.73	0.39
		Backward	0.21	0.37	0.43	0.42	0.31	0.84	0.35	0.25	0.23	0.38

2.5, our stochastic search strategy SHErLoCk described in Algorithm 1 was reasonably faster than the other ones and slightly dominant in accuracy as well. Moreover, it can be observed again that applying the stopping rule with the threshold STOP led to an enormous computational advantage for either search strategies with only a small drop in accuracy. For the sake

Table 2.5: Comparing the proposed search strategy (SHErLoCk) with other selection methods on binary classification problems of UCI datasets using an ensemble pool of $n = 100$ classifiers.

		Dataset (size)	MAGIC (19 020)	Spambase (4 601)	HIGGS (20 000)	EEG (14 980)	Musk (6 598)	Breast (699)	Mushroom (8124)	Gisette (13 500)	Adult (48 842)	Average
		Method										
Ensemble accuracy	MAXSTEP	SHErLoCk	94.66%	**97.88%**	**77.16%**	**96.98%**	**99.49%**	**98.97%**	**99.99%**	**94.16%**	87.88%	**94.13%**
		SA+	94.57%	97.79%	76.88%	**96.98%**	99.38%	98.95%	**99.99%**	94.03%	86.92%	93.87%
		SA	94.63%	97.75%	76.64%	96.11%	99.43%	97.98%	99.93%	93.94%	86.23%	93.63%
		Genetic+	**94.95%**	97.39%	76.41%	96.09%	99.43%	98.56%	99.89%	93.72%	**88.15%**	93.84%
		Genetic	94.79%	97.65%	76.84%	96.22%	99.08%	98.57%	99.91%	93.74%	87.19%	93.78%
		Pruning	93.88%	97.19%	76.63%	95.23%	98.71%	98.66%	98.94%	93.03%	85.79%	93.12%
	STOP	SHErLoCk	94.04%	96.88%	**76.73%**	95.64%	**99.28%**	97.61%	99.29%	93.19%	**86.99%**	**93.29%**
		SA+	**94.19%**	96.96%	76.02%	95.32%	99.18%	96.91%	98.97%	93.19%	86.01%	92.97%
		SA	94.17%	**96.99%**	76.02%	**96.09%**	99.21%	97.31%	99.03%	**93.44%**	86.02%	93.14%
		Genetic+	93.91%	96.91%	76.29%	95.96%	98.97%	**98.11%**	99.07%	93.41%	86.49%	93.26%
		Genetic	94.01%	96.46%	76.52%	95.88%	98.88%	97.77%	**99.90%**	93.23%	86.09%	93.19%
		Pruning	93.53%	96.38%	75.56%	95.19%	98.24%	97.12%	98.29%	93.01%	85.75%	92.56%
	DET	Forward	92.11%	94.92%	73.29%	95.14%	97.92%	96.69%	99.31%	92.98%	85.08%	91.94%
		Backward	93.92%	94.92%	75.11%	95.23%	97.92%	96.78%	99.31%	92.95%	84.83%	92.33%
Computational time	MAXSTEP	SHErLoCk	**194.12**	**206.89**	**191.91**	**203.88**	**214.28**	**201.03**	**186.67**	**159.32**	**178.01**	**192.90**
		SA+	349.62	290.89	390.82	278.56	253.59	375.71	313.56	301.25	311.87	318.43
		SA	251.02	291.13	269.22	278.39	269.59	228.44	286.23	259.92	258.62	265.84
		Genetic+	305.26	298.34	289.37	301.26	324.12	256.67	289.44	338.98	381.55	309.44
		Genetic	226.05	301.36	197.57	239.79	223.19	267.24	210.63	231.92	290.67	243.16
		Pruning	354.23	301.21	409.22	354.59	356.18	321.82	402.34	441.23	455.63	377.83
	STOP	SHErLoCk	**2.32**	**3.41**	**2.11**	**2.08**	**1.56**	**2.33**	**2.49**	**1.73**	**2.45**	**2.28**
		SA+	13.31	14.67	12.23	12.45	14.95	13.97	12.88	10.12	14.81	13.27
		SA	13.39	12.58	16.14	15.55	12.04	16.94	11.76	12.35	13.55	13.81
		Genetic+	12.56	12.61	15.95	13.37	13.49	16.68	14.88	13.14	12.59	13.92
		Genetic	13.53	13.58	12.88	15.47	11.51	13.67	12.02	11.27	12.28	12.91
		Pruning	19.41	11.78	17.32	36.36	21.69	31.55	22.92	15.34	19.53	21.77
	DET	Forward	0.59	0.72	0.81	0.92	0.69	0.89	0.93	0.96	0.94	0.83
		Backward	0.71	0.78	0.99	0.85	0.76	0.88	0.95	0.81	0.99	0.86

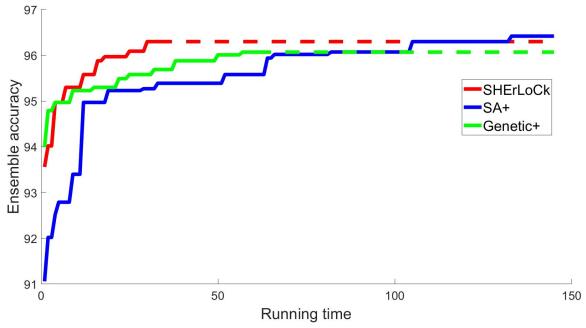
of completeness, we have also included the forward and backward selection techniques. As it can be observed from the tables, these deterministic techniques are naturally quicker than the others; however, their accuracies are reasonably low as well. Notice that, as deterministic techniques it is meaningless to limit the search time for them with either MAXSTEP or

STOP. Moreover, as for their 50% worst-case accuracy proved in section 2.2 we can see a better performance in our experiments.

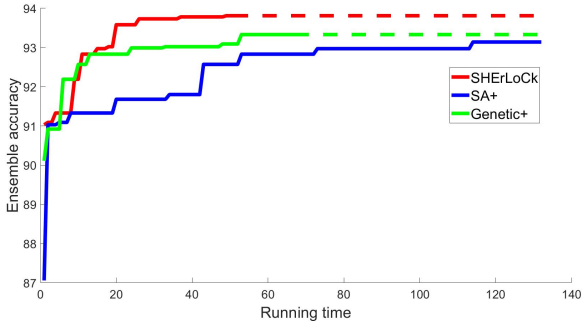
As for evaluating the accuracy of the ensembles of the trained classifiers, we have followed the next protocol. The individual classifiers have been trained according to the common guidelines with splitting all the datasets to training, cross-validation and testing parts. The accuracies p_i have been determined as their performances on the test set, since accuracy on the cross-validation subset does not reflect completely correctly this measure and these figures are available for only the tests set regarding the Kaggle competitors. As for ensemble accuracies gained by all the methods on the publicly available datasets, we have presented the experimental results for the test parts for Kaggle and for whole datasets for UCI. The reason is that official splitting is available only for the Kaggle datasets, but not for the UCI ones, so for future comparability whole dataset behavior seems more reasonable. Though this approach does not distort the relative behavior of the ensemble selectors, it led to a bit better performance. To address this issue correctly, we have checked the drop of accuracy if only the 30% randomly selected test parts of the datasets are included; for the results see Tables 4.2 and 4.3 in Appendix 4.10. We have found a small drop or a small rise regarding each ensemble creator method's performance on a particular dataset, so the trends shown in Tables 2.4 and 2.5 are not affected. The computational times naturally remain the same for both evaluation protocols.

We have also checked how the ensemble accuracies found by the different approaches increased regarding the elapsed time during the search. Notice that, the accuracies indeed have a monotonously increasing trend, since at each search step we store the best performing ensemble for all the approaches till the stopping condition is met. Our findings are depicted in Figure 2.2 (for $n = 30$) according to SHErLoCk, SA+, Genetic+ as best performing algorithms with indicating also the time points with dashed

lines when the respective methods were stopped. For this demonstrative analysis we have selected the two datasets EEG/Gisette requiring the largest/smallest computational times. For each of the analyzed approaches we can see sudden jumps coming from their stochastic behaviours.



(a)



(b)

Figure 2.2: The change of ensemble accuracy regarding the elapsed search time of the best performing approaches on the datasets EEG (a), and Gisette (b).

2.8 Investigating the extension of the proposed method to multiclass problems

The majority voting rule can be applied in a problem to aggregate the outputs of single object detectors in the spatial domain [25]; the votes of the members are given in terms of single pixels as candidates for the centroid of the desired object. In this extension, the shape of the desired object defines a geometric constraint, which should be met by the votes that can be aggregated. The practical example in [25] relates to the detection of a disc-like anatomical component, namely the optic disc (OD) in retinal images. Here, the votes are required to fall inside a disc of diameter d_{OD} to vote together. As more false regions are possible to be formed, the correct decision can be made even if the true votes are not in the majority, as in Figure 2.3. The geometric restriction transforms (2.1) to the following form:

$$q_{multi}(\mathcal{L}) = \sum_{k=0}^{\ell} p_{\ell,k} \left(\sum_{\substack{\mathcal{I} \subseteq \mathcal{L} \\ |\mathcal{I}|=k}} \prod_{i \in \mathcal{I}} p_i \prod_{j \in \mathcal{L} \setminus \mathcal{I}} (1 - p_j) \right). \tag{2.22}$$

In (2.22), the terms $p_{\ell,k}$ describe the probability that a correct decision is made by supposing that we have k correct votes out of ℓ. For the terms $p_{\ell,k}$ $(k = 0, 1, \ldots, \ell)$, in general, we have that $0 \leq p_{\ell,0} \leq p_{\ell,1} \leq \cdots \leq p_{\ell,\ell} \leq 1$.

The main challenge in solving this optimization problem is that the target function q_{multi} of the constrained majority voting is non-linear, non-separable. In general, Knapsack problems with these special kind of objective functions are investigated very rarely in the related papers, or only in that case when a strict restriction on their functional structure is given (e.g., the exponential type of target function is analyzed in [69]). That is, for a proper analysis we need some theoretical results for the optimization

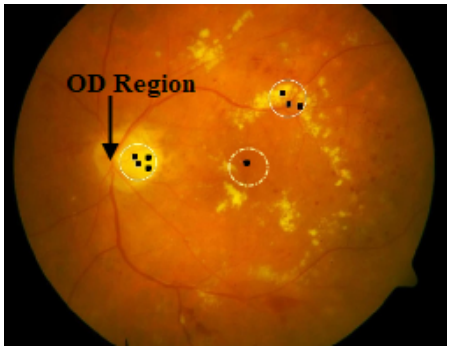


Figure 2.3: Successful OD detection with the same number of correct/false ensemble member responses.

of the specific target function (2.22) within the Knapsack framework.

The multiclass classification can be interpreted in a similar way as the binary one, just in case the prediction of the class for each element where it belongs to is made for three or more classes [70]. We encounter similar problems to find the optimal solution q_{multi} of multiclass Knapsack problem as in the binary case, but, besides the estimation of the behavior of the energy function, the terms $p_{n,k}$ need to be investigated, as well. It is reasonable to assume that the more classifiers out of the n ones give correct vote, the bigger probability $p_{n,k}$ for the good decision we get for the ensemble. Therefore, the terms $p_{n,k}$ are considered as values of a function F such that $p_{n,k} = F\left(\frac{k}{n}\right)$, where $F(\cdot)$ is a cumulative distribution function on $[0, 1]$.

We have the following theorem showing the behavior of the random variable q_{multi} (i.e. the expected ensemble accuracy and the variance), based on the random values of p_i-s.

Theorem 1. *Let $p \in [0, 1]$ be a random variable with $Ep = \mu$, $Var(p) = \sigma^2$, and p_i $(i = 1, 2, \ldots, n)$ are independent and identically distributed according to p. Furthermore let the energy function q_{multi} be defined by*

(2.22). Then for the expected ensemble accuracy $E(q_{multi})$ we have shown that

$$E(q_{multi}) = \sum_{k=0}^{n} F\left(\frac{k}{n}\right)\binom{n}{k}\mu^k\left(1-\mu\right)^{n-k}. \tag{2.23}$$

Furthermore, if n is large then

$$\sum_{k=0}^{n} F\left(\frac{k}{n}\right)\binom{n}{k}\mu^k\left(1-\mu\right)^{n-k} \sim \int_0^1 F\left(y\right)\delta\left(\mu\right)dy = F\left(\mu\right) \tag{2.24}$$

where $\delta\left(\cdot\right)$ is the Dirac function.

In case of large n, we have the variance of the ensemble accuracy

$$0 \leq \mathrm{Var}\left(q_{multi}\right) \leq F\left(\mu\right) - F^2\left(\mu\right) = F\left(\mu\right)\left(1-F\left(\mu\right)\right). \tag{2.25}$$

For practical issue, the following examples for the function F are important:

Arcsine law (distributed as Beta $(1/2, 1/2)$) with cumulative distribution function

$$F\left(y\right) = \frac{2}{\pi}\arcsin\left(\sqrt{y}\right), \quad y \in [0,1], \tag{2.26}$$

and Generalized Arcsine law (distributed as Beta $(1-\alpha, \alpha)$), as if the distribution of p is not known, then a Beta distribution is fitted to p.

From the results of Theorem 1 with respect to the expected value and the variance of the ensemble accuracy, the decision in the multiclass case for relatively large n is considered to be Bernoulli variated with parameter $F\left(\mu\right)$.

While the binary classification problem is closely related to the results of the binomial distribution, then in the multiclass classification the multinomial coefficients are supposed to have very important role in finding a formula for the values of $p_{n,k}$. As a first step, we simulated the multiclass classification problem for $c = 3$, $c = 4$ and $c = 5$ classes, by generating random numbers in $[0, 1]$, to decide which class is chosen. From the results of the simulations, we get approximate values for the terms $p_{n,k}$. In the next step, we give a closed formula for the values $p_{n,k}$, as well.

Let $c = d+1$ classes be given, where the $(d+1)$-th class has received k votes. Let the multinomial coefficients $b_{n-k,d}\left(x_1, x_2, \ldots, x_d\right)$ be given with $x_i \geq 0$, $\sum_{i=1}^{d} x_i = n-k$, $\underline{x} = (x_1, x_2, \ldots, x_d)$, where x_i is the number of votes for class i, and $\alpha_k\left(\underline{x}\right)$ is defined as the $|\{\underline{x} : x_i = k\}|+1$, where $|A|$ denotes the cardinaliy of a set A. Then for the terms $p_{n,k}(c)$ of accuracy in that case, we have

$$p_{n,k}\left(c\right) = \frac{1}{d^{n-k}} \sum_{0 \leq \underline{x} \leq k} \frac{b_{n-k,d}\left(\underline{x}\right)}{\alpha_k\left(\underline{x}\right)}, \tag{2.27}$$

where $0 \leq \underline{x} \leq k := (x_i | 0 \leq x_i \leq k, i = 1, 2, \ldots, d)$.

Formula (2.27) actually gives the number of ways that $n-k$ voters can vote independently of each other for the remaining d classes. It is also taken into account that if there are also $\alpha_k(\underline{x}) - 1$ non-preferred classes with k votes then in the case of a tie, there is a probability $\frac{1}{\alpha_k(\underline{x})}$ of choosing the right class.

For example, let us consider the following case: there are 4 classes and 10 voters in total. The preferred class has 3 votes and 2 non-preferred classes with 3 votes, then in this tie situation we choose the preferred class with probability 1/3 to make a good decision. Applying (2.27), we get the same results for the values of $p_{n,k}(c)$ in case of $c = 3$, $c = 4$ and $c = 5$ classes as before with the simulations.

In our experiments, the pool consists of eight OD detector algorithms with the following accuracy and running time values:
$\{(p_i, t_i)\}_{i=1}^{8} = \{(0.220, 31), (0.304, 38), (0.319, 34), (0.643, 69),$
$(0.754, 11), (0.765, 7), (0.958, 21), (0.976, 90)\}$ with $\sum_{i=1}^{8} t_i = 301$ secs. As we have mentioned earlier we can consider any resource type regarding the parameters t_i. For instance, we have considered the running times of the member algorithms here since some of them are not machine learning based ones. We can apply our theoretical foundation with some slight modifications to solve the same kind of knapsack problem for the variant

(2.22), transforming the model to reflect the multiplication with the terms $p_{\ell,k}$.

We have empirically derived the values $p_{8,k} = \{0, 0.11, 0.70, 0.93, 0.99, 1.00, 1.00, 1.00, 1.00\}$ for (2.22) in our task. To adopt our approach by following the logic of Algorithm 2, we need to determine a STOP value for the search based on μ_p and σ_p (calculated by (2.12)), and $\widehat{\ell}$ (calculated by (2.14)). However, since now the energy function is transformed by the terms $p_{\ell,k}$ in (2.22), we must apply Theorem 1 to derive the mean $\mu_{q_{\widehat{\ell}}}$ instead of (2.7) proposed in Algorithm 2. Accordingly, we had to find a continuous function $\mathcal{F}$ that fits to the values $p_{\ell,k}$, which was evaluated by regression and resulted in $\mathcal{F}(x) = b/(b + x^a/(1-x)^a)$ with $a = -3.43$ and $b = 101.7$, as also plotted in Figure 2.4. Now, by using Theorem 1, we have gained $\mu_{q_{\widehat{\ell}}} = \mathcal{F}(\mu_p)$.

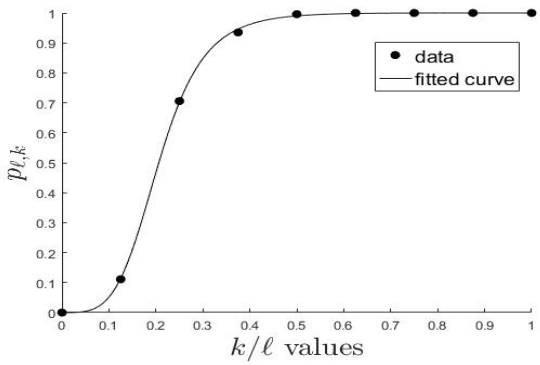


Figure 2.4: Determining the constrained majority voting probabilities $p_{\ell,k}$ for our OD detector ensemble.

For our experiment to search for the best ensemble, we have set the time constraint to be 80% of the total running time, with $T = 0.8 \sum_{i=1}^{8} t_i$. For this setup, we could estimate $\widehat{\ell} = 7$ and $\mu_{q_{\widehat{\ell}}} = 0.969$ for the expected ensemble size and mean accuracy, respectively. Then, these values have been considered for Algorithm 2 to demonstrate the performance of our

stochastic search method SHErLoCk with SA. As shown in Table 2.6, our search strategy outperformed SA also for the object detection problem both in accuracy and computational time. Because of the smallness of this setup, we have omitted a full experimental evaluation.

Table 2.6: Comparing SA with the proposed search strategy SHErLoCk on the OD detection problem.

Search method	Ensemble accuracy	Comp. time (secs)
	STOP	STOP
SHErLoCk	99.45%	0.07
SA	99.43%	0.16

2.9 Discussion

For the approximate number $\widehat{\ell_T}$ of the ensemble size, we have considered (2.14) when the member accuracy p is not a *beta* distribution. As an alternative notice that it is known that for a given λ, T and an independent exponential distributed τ_j, ℓ_T follows a Poisson distribution with parameter λT. We can use Lemma 3 and conclude that for a starting size of the ensemble, one may choose $\widehat{\ell_T}$ such that the remaining possible values are beyond the 5% error. It follows that we apply

$$\mathrm{P}\left(\ell_T > m_{0.05}\right) = 0.05, \tag{2.28}$$

where $m_{0.05}$ is the upper quantile of the Poisson distribution with parameter $T/\sum_{i=1}^{n} p_i$, or use the normal approximation to the Poisson distribution

$$\frac{\ell_T - T/\sum_{i=1}^{n} p_i - 0.5}{\sqrt{T/\sum_{i=1}^{n} t_i}} > 1.64, \tag{2.29}$$

which provides us the inequality

$$\ell_T > \frac{T}{\sum_{i=1}^{n} t_i} + 0.5 + 1.64\sqrt{\frac{T}{\sum_{i=1}^{n} t_i}} = \widehat{\ell_T}. \tag{2.30}$$

In our experiments we have used (2.14) instead of (2.30) to obtain $\widehat{\ell_T}$, since the latter provided slightly too large estimated size values. However, for other scenarios, it might be worth trying (2.30) as well.

As some additional arguments, we call attention to the following issues regarding those elements of our approach that might need special care or can be adjusted differently in other scenarios:

- We have assumed independent member accuracy behavior, providing solid estimation power in our tests. However, in the case of strong member dependencies, deeper discovery of the joint behavior might be needed.

- Stirling's approximation considered in (2.21) may provide values that are too small for the parameter MAXSTEP in the case of small pools. Since this is an escape parameter, a sufficiently large value should be selected in such cases instead. Note that on the datasets we examined, we found that it is not worth going above 15 000 iterations, because the ensembles' accuracies improved only slightly with higher step counts.

- The time profile $\lambda = 1 - p$ in section 2.6.1 is suited to our data; however, any other relationship between the member accuracy and time can be considered. In case other non-time-based resources are examined in the constraint, the exponential distribution can be changed accordingly. Nevertheless, the similar derivation of the estimation of the ensemble accuracy might be slightly more laborious depending on the selected distribution.

Chapter 3

Applications of ensemble methods in medicine

3.1 Predicting the Epidemic Curve of the Coronavirus (SARS-CoV-2) Disease Using Artificial Intelligence: An Application on the First and Second Waves

3.1.1 Introduction

The COVID-19 pandemic is considered a major threat to global public health. The aim of our study is to use the official epidemiological data to forecast the epidemic curves (daily new cases) of the COVID-19 using Recurrent Neural Networks (RNNs), then to compare and validate the predicted models with the observed data.

The initial epidemic curves of the COVID-19 outbreak from Hubei, China showed a mixed pattern, indicating that early cases were likely from a continuous common source e.g., from several zoonotic events in Wuhan, followed by secondary and tertiary transmission providing a propagated source for the later cases [86]. The propagated (or progressive source) epidemic curve visualizes the spread of an infectious agent that may be transmitted from human to human starting from with a single index case, that continues to further infect other individuals. This shows up as a series of peaks on the epidemic curve that starts with the index case, followed by successive waves of the infection set apart with respect to the incubation period of the pathogen. The waves continue to follow each other until appropriate mitigation measures, prevention, or treatment are implemented, or the pool of the susceptible population becomes infected. This is a theoretic curve, that is generally influenced by lots of other factors [86].

Various mathematical models may demonstrate and predict the dynamics of different infectious diseases [36]. These models, used to simulate the dynamics of infectious diseases, may be based on statistical, mathematical, empirical, or machine-learning methods [71]. One class of AI, a form of artificial neural networks, the Recurrent Neural Networks (RNNs) with Long Short-Term Memory (LSTM) were previously used to model and forecast the influenza epidemic, with strong competitiveness and reliable results [87].

During the COVID-19 pandemic, various simulation studies reported the use of different AI-based methods to forecast the projections of the COVID-19. In [21], an LSTM algorithm with ten hidden units was used to predict the spread of the COVID-19 in terms of confirmed cases and deaths in six gulf countries. In India, a data-driven model based on LSTM was used to predict cases and recoveries, considering the imposed governmental preventive measures like lockdown and isolation [83]. Furthermore, in [52], the prediction of new cases of COVID-19 in several European countries was discussed using three approaches, namely, Auto-Regressive Integrated Moving Average (ARIMA), Nonlinear Autoregression Neural Network (NARNN), and Long-Short Term Memory (LSTM)
.

The results presented in this section are published in my following publication: [46].

3.1.2 Datasets

We used the publicly available datasets from the WHO and Johns Hopkins University for the following countries to create the training dataset: Austria, Belgium, China (Hubei), Czech Republic, France, Germany, Hungary, Iran, Italy, Netherlands, Norway, Portugal, Slovenia, Spain, Switzerland, United Kingdom (UK) and the United States of America (USA) [42]. Given that most infected people in China were from Hubei province, only data from that province was included. For each country, the date of the first reported infection was set as Day 1 for the disease time scale. (Figure 3.1).

Several important conditions had to be taken into account when designing the data set for the first wave. When determining the date of the first illness (first identified case), point-source outbreaks were omitted (e.g., those cases where single verified cases were isolated, and no further

transmission has occurred). This was important to avoid distortion of the propagated epidemic curves. In Belgium, for example, the first illness occurred on February 02, 2020, and there were no further cases reported for up to 26 days. The next illness occurred on March 01, 2020. The inclusion of the early case from February would contribute to a false learning rule for the AI, hence corrupting the results. As for Hubei Province, the first officially available data was on January 22, 2020. This cannot be considered as the first day of the illness, thus the first infection was arbitrarily defined to occur on January 01, 2020. To account for the extreme variability of daily incident cases reported which probably reflects delays in reporting procedures, a moving average was used (covering 3 days) for the Hubei dataset.

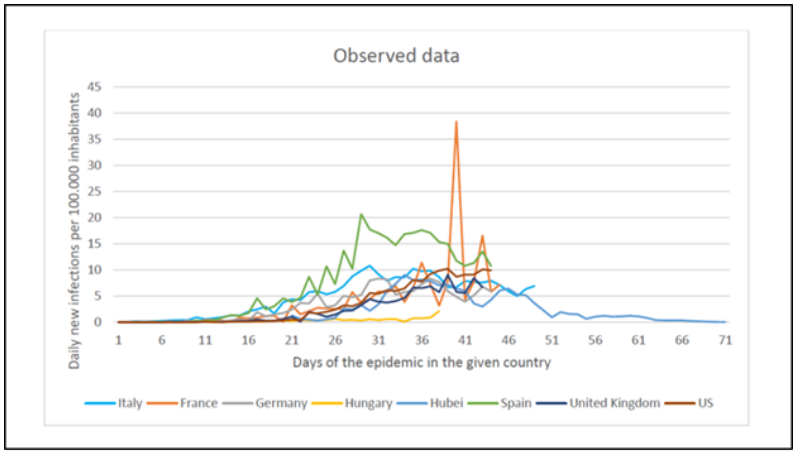


Figure 3.1: The Historical Datasets of Different Countries From First Pandemic Wave

Accordingly, an epidemic curve was obtained for each country with a time series where the first day denotes the day of the first confirmed case, and each successive day indicating the number of newly confirmed cases that day. To account for the country-specific differences in the size of the population, the number of daily new cases was normalized for 100.000

inhabitants in each country. The observation period varies for each country, given the difference of time elapsed since the disease initiation in that country. Accordingly, the longest time series covers the observation period of 90 days (in case of the first wave). e.g., in Hubei, with the first 22 days lacking valid data and the next 68 days having data. The shortest observation period was in Slovenia with only 30 days.

For the first wave, the training data set was obtained by averaging the daily incidence rates per 100 000 inhabitants across the 17 countries included, for each day in the time series. When calculating the average, missing data was left blank, i.e., NULL, e.g., countries that did not contain data for a specific day were excluded from the calculation of average. The resulting training data set is shown in Figure 3.2. It should be noted that the first part of the data set (up to the initial 30 days since Day 1 of the epidemic) contains data for almost all the countries listed, whereas the end of the data set contains only data from Hubei (Figure 3.2).

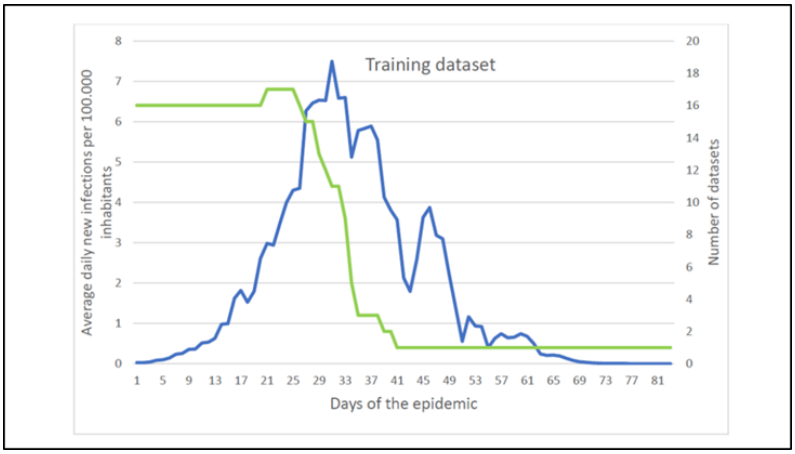


Figure 3.2: The training dataset. Average daily new infections per 100.000 inhabitants (line blue) and the Number of datasets (line green)

In order to test our model more accurately, we also examined the second waves' data. To obtain more accurate results for the second wave

Table 3.1: The Day 1. of the Second Wave.

Country	Day1
France	2020.09.13
Germany	2020.10.23
Italy	2020.10.13
Spain	2020.09.13
Hungary	2020.10.13
UK	2020.09.13
Spain	2020.10.13

we have created an interconnected neural network model, whose first part is the base RNN trained on the first wave data. The second part of the extended model is the neural network component trained on the second wave dataset. The second wave data for each country under study consisted of 85 days. Of these, the first 60 days were used for retraining and the next 25 days for prediction and validation. The training datasets used per country for the second wave are presented in Figure 3.3. However, for each country, the course of the pandemic is different, so the first day of the second wave is determined by country. The first day of the second wave in each country is shown in Table 3.1.

3.1.3 RNNs-Based Models for Prediction

The state-of-the-art for time series analysis is AI-based analytic tools, which have the best prediction performance. Recurrent Neural Networks (RNNs) are specifically designed to cope with sequential input, characteristic of textual or temporal data. This architecture is a neural network-based architecture, that contains hidden layers chained according to the time step, with a possibility to predict the next sequence element(s). A

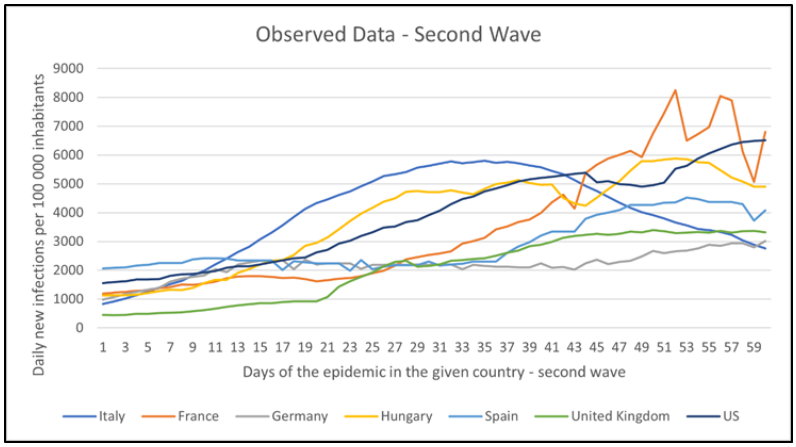


Figure 3.3: The Observed Datasets of Different Countries From Second Pandemic Wave

time series has a special temporal form, where the input to the i-th hidden layer is at the i-th time-step that has a corresponding $x(i)$ observation. In its original form, a simple RNN tries to predict the next sequence element, however, for the purpose of the current analysis, an encoder-decoder variant is a more natural choice, similarly, to machine translation [74]. For our specific scenario this means that during the encoder phase including time steps $1, \cdots, t$ the RNN is fed with the already known time series data (the average of the number of new cases normalized to 100 000 inhabitants for day $1, \cdots, t$, respectively), followed by prediction in the decoder phase for the future time steps $t+1, \cdots, T$. In our analysis, $T = t + 1 = 90$ days is the longest known (Hubei) time interval. Since this covers quite a long data sequence, we have used gated recurring units (namely Long Short-Term Memory – LSTM units) [39]. Figure 3.4 depicts our basic RNN architecture showing how unknown time series elements are predicted. Figure 3.4 also shows how the information collected in the first t time-steps are aggregated with a fully connected (dense) neural network layer and a consequent regression output layer to determine a predicted

number of new patients as $x(t+1)$. We used our own approach to design the architecture and build the encoder-decoder process according to the problem. The construction of the basic network begins with a Sequence Layer, followed by the LSTM blocks shown in Figure 3.4, which have a memory capability for the previous state. With this feedback process, the prediction can get much closer to the real one. Dropout layers were added to the LSTM layers of the network to control overfitting. Dropout is a regularization method where input and recurrent connections to LSTM units are probabilistically excluded from activation and weight updates while training a network. It has the effect of reducing overfitting and improving model performance.

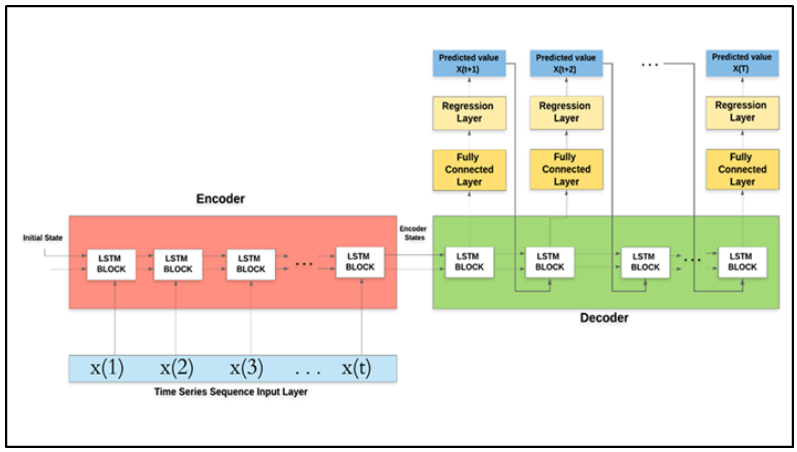


Figure 3.4: The basic Recurrent Neural Network (RNN) Architecture used for Prediction.

We experimented by gradually increasing the epoch number from 50 to 300. The best results were obtained after 150 epochs. In later epochs, there was an inconsistency in both machine capability and the accuracy score. To save training time, we have implemented mini-batch gradient descent in the training process, and the batch size we used was 8. To optimize the training process, we took advantage of the ADAM optimizer by

setting the learning rate at 1e-4 and reduced it by 1e-6 for each subsequent epoch.

As for adjusting the hyperparameters (number and components of LSTM layers, dropout probability, optimizer, mini-batch size, learning rate) of our neural model we have applied the Bayesian algorithm, which is well suited for optimizing hyperparameters of classification and regression models.

During the evaluation of the results, we have used this trained basic model, but for each country, the state of the basic model was updated with the help of the training data set of that country.

In predicting the second wave, we had much more metadata available, such as viral replication rates, mortality data, numbers indicating the extent of restrictions imposed by governments. Adding these extra features to the system we have developed a solution that takes better account of the circumstances in the prediction, so we can get a more accurate prediction of the number of new cases per day. For the second wave prediction, we have created an interconnected neural network model, whose first part is the base RNN trained on the first wave data. The second part of the extended model is the neural network component trained on the second wave dataset and augmented with the metadata mentioned above. After these two components were connected following their training, they have undergone a state update, which consisted of a retraining step regarding the specific country data to be predicted. The essence of the connected model is that the states of the two sub-networks are updated simultaneously for a given country and the final decision is reached as the weighted sum of the outputs of the two networks. These weight parameters are also embedded in the interconnected neural architecture so adjusted automatically during the training process.

To assess the possible specificities regarding the countries, two approaches were used for prediction as follows:

- Prediction 1: An algorithm to update the training step and subsequent prediction was formulated. This updating step is based on the general recommendations of transfer learning that considers the already known time interval for the given country and re-training is done in small increments of the RNN network accordingly [74]. Thus, we start predicting the first unknown element $x(t+1)$ from the last 5% of the known data, and the same principle is applied to each subsequent element. Moreover, after each prediction steps our RNN architecture was re-trained, and the subsequent elements were predicted with this updated RNN.

- Prediction 2: We start predicting the first unknown element $x(t+1)$ from the last known $x(t)$, and all the subsequent elements are predicted only from the preceding ones. Here the rules depicted from the training data set are used, no retraining occurs.

The intuitive interpretations of the difference between Prediction 1 and Prediction 2 are as follows. Prediction 2 makes its predictions utilizing the information derived from the training data set, reflective of the trends in the average time series. It follows those predictions will comply primarily with the Hubei time series, especially in the far future. Therefore Prediction 2 shows the highest fidelity to the country-specific future scenario if the approach to mitigate the epidemic is similar to that in Hubei. Accordingly, this scenario is also reflective of a country-specific future state given the practices of Hubei were followed in said country. On the other hand, Prediction 1 is yielded after the neural network is retrained after any prediction, providing more valid insight into what is expected if the country goes on with the mitigation practices seen during the observation period. This intuition can be also used for the evaluation of the second wave of the pandemic because in this case, the prediction architecture includes the neural network which was trained during the first wave.

3.1.4 Validation

For the learning dataset, we used the data from the first and second pandemic wave. For the first wave, the available factual data is taken from the first case reported in a country until April 10, 2020. For the second wave, the start dates shown in Table 3.1 were used to define the dataset. Based on that, we have made the above-mentioned two predictions (1 and 2). For the first wave, validation was done using the second half of the factual data (these data was not part of the training dataset), while for the second wave, the last 25 days were used. The amount of Root Mean Squared Logarithmic Errors (RMSLE) was used for validation. In our analysis, the possible bias regarding the different ratios between the observed and predicted values are interpreted using the RMSLE. Let n be the number of days used for validation. Let p_{1i} and p_{2i} be the number of new cases per day obtained using the two prediction methods in the examined time interval and let a_i be the actual data for the given days. Err_1 and Err_2 will be RMSLE for Prediction 1 and Prediction 2, respectively, where:

$$Err_i = \sqrt{\frac{1}{n}\sum_{j=1}^{n}(\log(p_{ij}+1) - \log(a_j+1))^2}, \tag{3.1}$$

and $i \in \{1, 2\}$.

We have calculated also the root Mean Square Error (RMSE) and the Mean Absolute Percentage Error (MAPE), as follows:

$$RMSE_i = \sqrt{\frac{1}{n}\sum_{j=1}^{n}(p_{ij} - a_j)^2} \tag{3.2}$$

$$MAPE_i = \frac{100}{n}\sum_{j=1}^{n}\left|\frac{p_{ij} - a_j}{a_j}\right| \tag{3.3}$$

where $i \in \{1, 2\}$.

3.1.5 Results

This section shows the outcomes for Prediction 1 and Prediction 2 of the individual country-level data for France, Germany, Hungary, Italy, Spain, the United Kingdom (UK), and the United States of America (USA) (Figures 4.6-4.19). In each graph, the first day represents the first illness/case of each country. The yellow line represents the factual data for 85-91 days from the first illness/case for both waves. The values obtained by the two prediction methods for each country are represented by blue and green lines. The blue line shows Prediction 1, and the green line shows Prediction 2. For each main graph, the small graph in the upper right corner contains the daily error values calculated for the predictions. The more accurate the prediction, the smaller the RMSLE error. It should be noted that if the error function is parallel to the x-axis, it means that the trend of the prediction is the same as the real trend, only at a lower or higher scale. Also, total Root Mean Square Error (RMSE), Root Mean Squared Logarithmic Errors (RMSLE), and the Mean Absolute Percentage Error (MAPE) by country are shown in Table 3.2 and 3.3. The large differences in the values of the metrics used for the evaluation in the two waves studied may be explained by the emergence of new virus mutations and by the fact that the population did not take the restrictions as seriously as in the first wave.

3.1.6 Conclusions

We used publicly available datasets from the World Health Organization and Johns Hopkins University to create a training dataset, then we employed RNNs with gated recurring units (Long Short-Term Memory - LSTM units) to create two prediction models. Our proposed approach considers an ensemble-based system, which is realized by interconnecting several neural networks. To achieve the appropriate diversity, we froze

Table 3.2: Total Root Mean Square Error (RMSE), Root Mean Squared Logarithmic Errors (RMSLE), and the Mean Absolute Percentage Error (MAPE) for the first wave.

Country	RMSE of Pred1	RMSE of Pred2	Mean of RMSLE of Pred1	Mean of RMSLE of Pred2	MAPE of Pred1	MAPE of Pred2
Hungary	0.31	0.42	0.06	0.107	51.9	66.5
UK	3.12	4.56	0.234	0.455	43.35	65.75
Italy	1.56	1.83	0.114	0.155	20.89	28.5
Spain	4.23	3.96	0.266	0.181	137.41	74.65
Germany	1.51	1.49	0.147	0.108	94.18	52.39
France	8.34	5.54	0.513	0.307	585.07	189.7
USA	3.48	5.78	0.216	0.528	35.52	63.86

Table 3.3: . Total Root Mean Square Error (RMSE), Root Mean Squared Logarithmic Errors (RMSLE), and the Mean Absolute Percentage Error (MAPE) for the second wave.

Country	RMSE of Pred1	RMSE of Pred2	Mean of RMSLE of Pred1	Mean of RMSLE of Pred2	MAPE of Pred1	MAPE of Pred2
Hungary	597.47	640.46	0.097	0.099	24.94	25.62
UK	897.02	1511.89	0.117	0.311	22.34	50.87
Italy	216.64	162	0.033	0.024	7.24	5.65
Spain	488.98	467.48	0.07	0.064	17.05	15.24
Germany	349.79	409.15	0.057	0.064	13.07	13.45
France	698.1	621.62	0.094	0.081	23.18	19.32
USA	1216.21	3188.18	0.088	0.308	18.13	0.11

some network layers that control the way how the model parameters are updated. In addition, we could provide country-specific predictions by transfer learning, and with extra feature injections from governmental constraints, better predictions in the longer term are achieved. We have calculated the Root Mean Squared Logarithmic Error (RMSLE), Root Mean Square Error (RMSE), and Mean Absolute Percentage Error (MAPE) to thoroughly compare our model predictions with the observed data.

We reported the predicted curves for France, Germany, Hungary, Italy, Spain, the United Kingdom, and the United States of America. The result

of our study underscores that the COVID-19 pandemic is a propagated source epidemic, therefore repeated peaks on the epidemic curve are to be anticipated.

The approach we have proposed provides a much more realistic prediction over a longer period than other work in the literature [9, 57, 67]. By optimizing classical recurrent neural network models, adding extra features, and combining transfer learning with a complex architecture of interconnected subnetworks, we can predict the entire epidemic curve of a given wave of an epidemic with good approximation accuracy based on a few weeks of data from the outbreak. Besides, the errors between the predicted and validated data and trends seem to be low.

However, the emergence of different viral mutations also changes the behavior of the epidemic curve, for which the presented neural network model is not fully prepared yet. This is because the behavior of the training dataset strongly influences the prediction behavior. Our plans include improving this shortcoming of our model. Since the parameters of the mathematical models describing epidemic spread are easily updatable, we can use different mathematical approaches (e.g., SEIR) to simulate the epidemic spread process by considering the occurrence of multiple mutations. The outputs of these simulations are then used as a training data set to further develop the neural network model. The validation process will be based on the effects of currently available COVID-19 virus variants (e.g., the British or the Indian mutations). Thus, the overall future goal is to develop a much more flexible prediction model.

3.2 Detecting outlier and poor quality medical images with an ensemble-based deep learning system

3.2.1 Introduction

In recent years, deep learning has dramatically improved the state of the art technology in many research areas, including computer vision, speech recognition, document recognition and natural language processing [62, 64, 55]. Deep convolutional neural networks have made a breakthrough in the processing of images, video, speech and audio; and ones have reported a new tool for solving visual recognition problems, such as image classification, semantic segmentation and object detection [64, 90, 34, 32]. Convolutional neural networks (CNNs) offer an effective architecture for extracting highly relevant statistical samples of large datasets.

Because CNN is able to learn from a large dataset, it is important to filter out the false patterns generated during data collection from our initial database. This procedure is called outlier detection [12]. There are many applications for this procedure in other areas, such as network effects (for example, an anomalous traffic pattern on a computer network when a hacked computer sends sensitive data to an unauthorized destination), detection of credit card or telecommunication fraud (a sudden change in usage pattern can mean fraudulent use, such as stolen cards/phone conversations) [68].

We introduce a hybrid outlier detection method by combining convolutional neural network with SVM which improves the performance of CNN. Furthermore, we can construct an ensemble when these hybrid classifiers are considered as ensemble members. We will show that this ensemble system successfully outperforms each individual hybrid CNN-

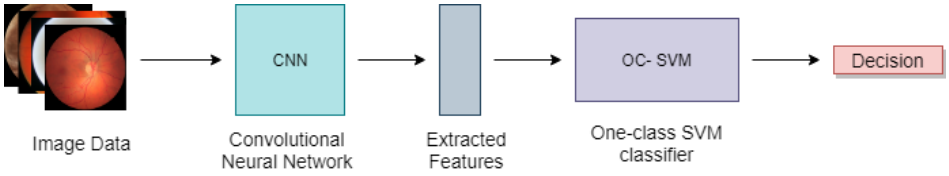


Figure 3.5: The hybrid CNN-SVM model: CNN combining with one-class SVM classifier

SVM method, as well.

The results presented in this section are published in my following publication: [79].

3.2.2 The hybrid CNN-SVM method

One-class Support Vector Machines are widely used to identify anomalies. A Support Vector Machine (SVM) is a discriminative classifier where the best separating hyperplane is to be found. It means that we search for the hyperplane which separates all data points into two classes with the largest margin between them. In two dimensional case, this hyperplane is a line dividing a plane in two parts for the two classes. We remark that an SVM can be used in multinomial classification problems, as well. In this case we can consider a one vs. all classification, in which the class with the highest score is separated from all the other classes together.

A convolutional neural network (CNN) consists of hidden layers for parameters to be learnt. In classification problems, this method gives the correspondence in which one of the given class labels is mapped to each input data. Conventionally, the Softmax function is applied as classifier in the last layer of CNN.

Transfer learning [3] is commonly used in deep learning applications, because fine-tuning a network with this technique is usually much faster than training it from scratch with randomly initialised weights, and it re-

quires a smaller number of training images, as well. To retrain these models for a specific task, it is necessary to replace their final layers with ones that are adapted to match our specific classification problems and datasets. Support Vector Machine can be used as an alternative of Softmax function for classification.

In the proposed method, we create hybrid models using the CNN features extracted from the hidden layers of pre-trained models as input of one-class SVM with the classic linear kernel to train the classifier. Fig. 3.5 shows the workflow of the hybrid CNN-SVM model for classification. The experimental results show that the hybrid CNN-SVM methods improve the performance of the classic CNN methods.

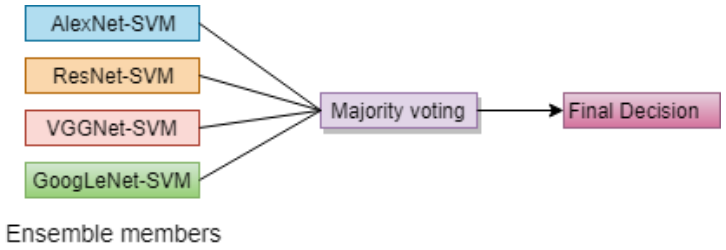


Figure 3.6: Ensemble system of hybrid CNN-SVMs for final decision

Ensemble of the hybrid CNN-SVMs

In numerous application fields, ensemble creation is a well-known and popular approach to raise the performance of individual methods [61]. We can encounter with such technique in pattern recognition applying neural networks, as well. To make the final decision, some kind of voting rules (e.g. the classic or weighted majority ones) is applied to aggregate the individual decisions of the ensemble members.

In our proposed method, the hybrid CNN-SVM methods $S_1, S_2, \ldots, S_n$ $(n \geq 1)$ are considered as the members of the ensemble system S. Let

$S_i(x)$ $(1 \leq i \leq n)$ and $S(x)$ denote the classification result of the ensemble member S_i and the ensemble S, respectively. In the fusion model of the hybrid networks, classic majority voting is applied. The final class label for the input data is derived as the one provided by the majority of the individual hybrid CNN-SVMs. In Fig. 3.6, the workflow of the ensemble system constructed from four different hybrid CNN-SVM models for classification is presented.

In binary classification problems, when class labels are 0 and 1, we can get the final decision of ensemble S with n members for the input data x by applying majority voting as follows:

$$S(x) = \begin{cases} 1, & \sum_{i=1}^{n} S_i(x) > \lfloor \frac{n}{2} \rfloor + 1 \\ 0, & otherwise \end{cases} \tag{3.4}$$

In this case, each CNN-SVM is forced to assign a single class label 0 or 1 to x, and these votes are aggregated by (3.4).

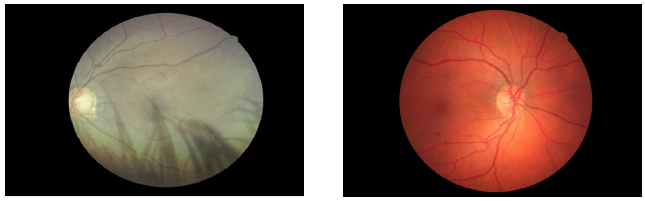
Datasets and preprocessing

In order to train the CNNs to detect the outliers for retinal images, we apply a Kaggle database[1] which consists of 88702 retinal images. The dataset contains about 65% images of good quality and 35% outlier images. The majority of the inappropriate images in the database are not in proper quality; e.g. blurred, out of focus or the retina itself is not inside the ROI. Furthermore, those images are taken in consideration, as well, in which other objects than the retina are presented, e.g. eyelashes because of the patient's blinking while taking the picture.

Fig. 3.7 shows example of retinal images from the database with poor (left) and good quality (right).

The other database of skin lesion images is set up by ourselves; with the help of a dermatoscope, which is a clip-on tool with a magnifier lens

[1] https://www.kaggle.com/c/diabetic-retinopathy-detection/data



(a) (b)

Figure 3.7: Poor quality image, the eyelashes cover some part of the retina image (left). Good quality image, retina can be detected properly (right)

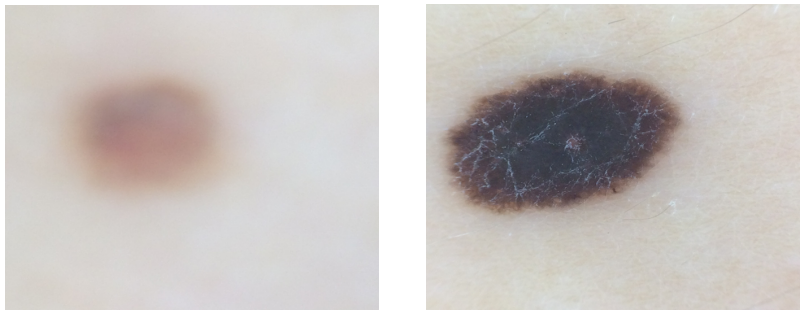
for cellphones and is able to take high quality images. More precisely, we have acquired skin images with a dermatoscope FotoFinder Handyscope attached to an Apple iPhone 6 mobile phone as our primary aim is to support home health-care, where outlier images – random or poor quality ones – may occur much more frequently than under clinical conditions.

Among these images we happened to encounter ones that contained errors; such as blurred ones or flares. In order to collect images, we worked with 60 subjects. We needed to annotate these two databases, which meant the labelling of the images whether they represented appropriate or outlier samples. In this way, we have collected 2000 skin lesion images, 2000 non-domain and 2000 poor quality ones with the latter two labelled as outliers in our dataset. An example of skin lesion image with poor (left) and good quality (right) is presented in Fig. 3.8.

Table 3.4: Experimental classification results on the skin lesion test set.

Measures	AlexNet	S_1	ResNet	S_2	VGGNet	S_3	GoogLeNet	S_4	Ensemble	Ensemble of hybrid models
ACC	0.898	0.968	0.893	0.967	0.915	0.964	0.917	0.971	0.931	0.979
AUC	0.901	0.979	0.910	0.976	0.920	0.973	0.911	0.976	0.948	0.986
REC	0.897	0.959	0.891	0.951	0.919	0.957	0.902	0.955	0.921	0.968
SP	0.915	0.978	0.909	0.969	0.907	0.954	0.923	0.975	0.937	0.978
PRE	0.913	0.973	0.911	0.964	0.917	0.965	0.921	0.968	0.929	0.976

After the annotated data collection step we have been able to set up



(a) (b)

Figure 3.8: Poor quality image, the skin lesion is blurred (left). Good quality image, the skin lesion is properly detected (right)

Table 3.5: Experimental classification results on the retinal image test set.

Measures	AlexNet	S_1	ResNet	S_2	VGGNet	S_3	GoogLeNet	S_4	Ensemble	Ensemble of hybrid models
ACC	0.966	0.984	0.953	0.976	0.969	0.974	0.964	0.976	0.978	0.991
AUC	0.977	0.981	0.971	0.979	0.974	0.973	0.979	0.984	0.981	0.988
REC	0.967	0.969	0.950	0.963	0.971	0.969	0.966	0.967	0.977	0.989
SP	0.961	0.987	0.963	0.978	0.970	0.974	0.965	0.979	0.971	0.982
PRE	0.963	0.978	0.966	0.969	0.967	0.965	0.961	0.968	0.969	0.979

the training and test dataset. It holds great importance for us to train the CNN in such manner to be capable of filtering out anomalies and errors. As for learning, we have considered 70% of the cases for training and the remaining 30% for testing purposes.

Measures for performance comparison

We introduce the performance measures which are considered in our classification problems to compare the efficiency of the classic CNNs, the hybrid CNN-SVMs and the ensemble system of the hybrid CNN-SVMs.

These methods can be evaluated according to the overall score on the test set calculated as the average of the area under the receiver operating characteristic curve (AUC) corresponding to classification results. Fur-

thermore, we calculate the frequently used performance measures: accuracy, precision, specificity and recall.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}, \tag{3.5}$$

$$Precision = \frac{TP}{TP + FP}, \tag{3.6}$$

$$Specificity = \frac{TN}{TN + FP}, \tag{3.7}$$

$$Recall = \frac{TP}{TP + FN}, \text{ where} \tag{3.8}$$

TP (true positive) is the number of positive cases predicted correctly by the classification model, and the FP (false positive) is the number of negative cases that the classification model predicts incorrectly as positive. TN (true negative) is the number of negative cases predicted correctly by the classification method, and the FN (false negative) denotes the number of positive cases that the classification model predicts incorrectly as negative.

3.2.3 Experimental results

In our experiments, we used the following CNNs: AlexNet [48], VGGNet [72], ResNet [34] and GoogLeNet [76].

AlexNet is an 8 layers deep CNN with an image input size of 227×227 pixels. VGGNet, ResNet and GoogLeNet are 19, 50 and 22 layers deep respectively, and they have a common input image size of 224×224 pixels. For this reason, we have resized the training and test images to the appropriate sizes before they were used as input of the networks.

The aforementioned networks are available as models pre-trained on the database ImageNet [1], which contains more than one million images

representing 1,000 classes. As a result, the networks have learned rich feature representations for a wide range of images.

In CNNs, deeper layers contain higher level image features, that are constructed using the lower level ones of the earlier layers. In our approach, we extract the learned image features from the pre-trained CNNs, and use them as features to train an SVM image classifier. Namely, we replaced the last fully connected layers and the classification one of pre-trained networks. During the training and the validation process, a single fold was applied for training/testing separation by splitting the dataset randomly: 70% for training and 30% for testing. We use different batch sizes for different models. We used a batch size of 32 and a batch size of 64 to facilitate the learning process and to save training time. The training process lasted 8, 15 and 21 epochs for the different models to avoid overfitting. We considered Cross Entropy as the loss function of our model at 10^{-2}, 10^{-4} and 10^{-5} learning rates in minimizing the loss of the model produced. After that, the features from the last (fully connected) layers are used to obtain the feature representations of the training and test images, these features are fed into the SVM classifier. We applied k-fold cross validation ($k = 10$) to check the classifier accuracy.

From each of the above CNNs, we are able to set up a hybrid neutral network CNN-SVM by using transfer learning and feature extraction methods. In this way, four hybrid algorithms (S_1: AlexNet-SVM, S_2: VGGNet-SVM, S_3: ResNet-SVM, and S_4: GoogLeNet-SVM) are constructed. We can compare the efficiency of the hybrid CNN-SVM methods to the classic CNNs in classification problems. We used the performance measures introduced in previous section for the comparison. The classification results of the CNNs and the CNN-SVMs in the outlier detection for skin lesion images are presented in Table 3.4 The results show that each hybrid CNN-SVM algorithm outperforms the corresponding CNN method. In the last two columns of Table 3.4, the classification results of

the ensemble of the classic CNNs and of the hybrid CNN-SVM methods are presented. From these results, we can observe that the ensemble creation technique improves the performance of the member algorithms in both cases.

Very similar observations can be made as in the previous case, for the comparison of the efficiencies of the considered algorithms for the retinal image dataset, as well. The classification results in the outlier detection for retinal images are presented in Table 3.5.

To summarize the experimental results, we get that the ensemble of the hybrid CNN-SVMs performs the best in comparison of the other methods in both classification problems for filtering the anomalies. Some classification results of this ensemble method in outlier detection in retinal and skin lesion images are shown in Fig. 3.9 and 3.10, respectively.

3.2.4 Conclusions

Our aim was to develop a procedure that is able to efficiently filter images that exhibit different kind of anomalies in a dataset. As specific applications of outlier detection, we considered colour fundus images and skin lesions in dermatoscopic images. Our experimental results show that each hybrid CNN-SVM method has turned out to be more accurate than the corresponding CNN in filtering images with anomalies. Furthermore, we have experienced that the proposed ensemble of the hybrid CNN-SVM methods outperforms its members in eliminating the outliers from the databases. The decision-making ensemble based on the proposed hybrid CNN-SVM method may serve as a pre-filtering component integrated into complex medical decision support systems.

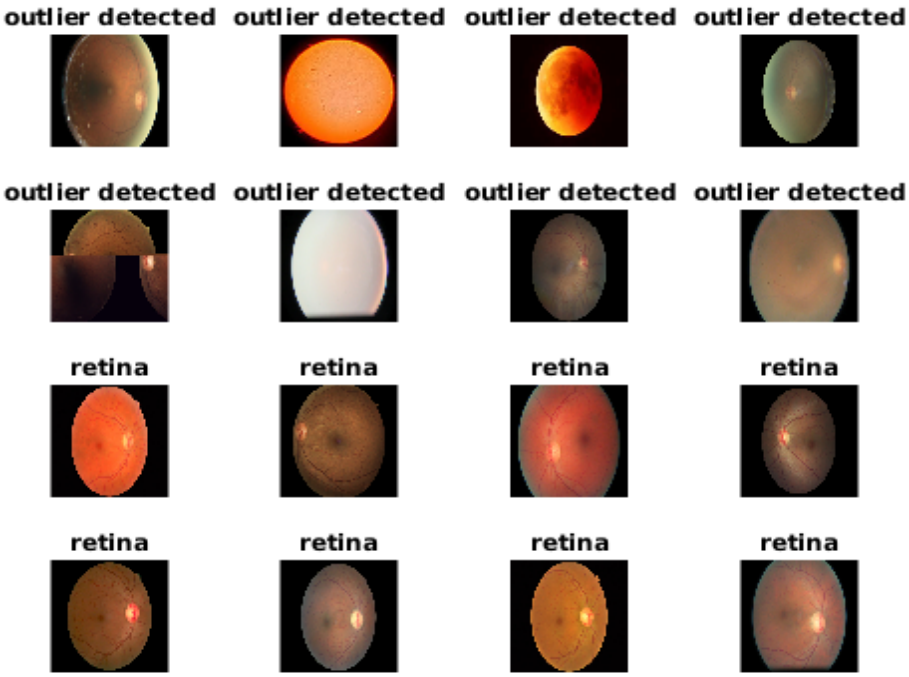


Figure 3.9: Most normal and in-class anomalous retinal images detected by ensemble of classifiers.

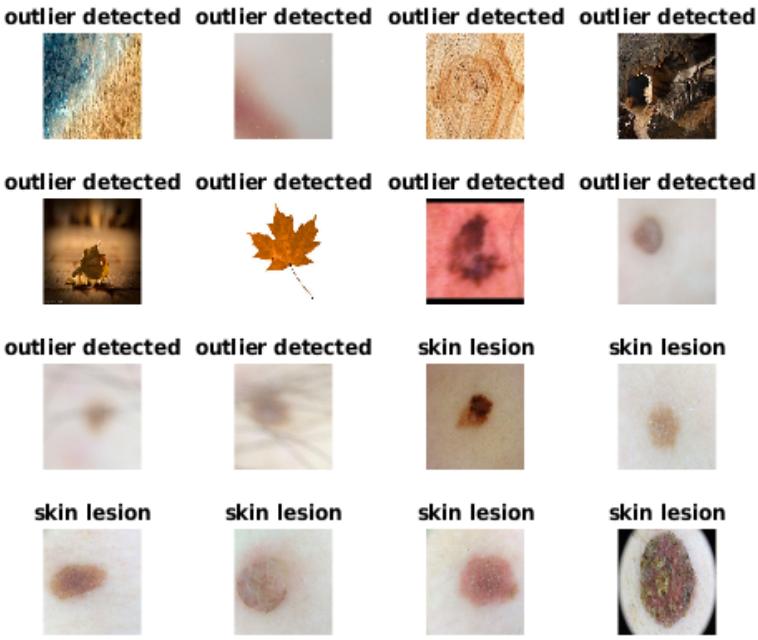


Figure 3.10: Most normal and in-class anomalous skin lesion images detected by ensemble of classifiers.

Chapter 4

Deterministic methods for measuring pattern regularity

4.1 Efficient Texture Regularity Estimation for Second Order Statistical Descriptors

4.1.1 Introduction

Examination of textures has received a strong attention form researchers for many years. It is important in several cases to determine whether the texture is regular, or to determine at least the presence of some kind of regularity of the pattern within the texture. Analyzing the regularity of patterns has importance in several fields. Checking pattern regularity is widely used e.g. in medicine. For example, examination of the regularity of the pigment network has an important role in recognizing typical/atypical lesion behavior [28]. We can also mention its use in analyzing the regularity of EEG data on patients suffering from Alzheimer's disease [2]. Beyond clinical fields, this approach is highly popular in other image processing related tasks like detecting possible regularities of image

structures [24], human movement patterns [23], or monitoring eye movements/recognizing gestures [65]. In general, regularity measure is very often used in classification problems, or to select specific information from the data.

Among texture analysis approaches, a popular one is to consider first or second-order statistical descriptors based on the intensity histogram. As a second-order method, the analysis of measures obtained from the co-occurrence matrix is widely considered [63]. To generate a co-occurrence matrix, we need a position vector, after which the matrix entries accumulate the relative occurrence of pixel pairs of given intensities residing at the end points of the position vector. The position vector is usually considered to be as a short one, however, no explicit recommendation for its length is available in the literature. The current empirical application scheme is to rotate the vector at angles of 45, 90, 135 degrees, and create co-occurrence matrices for each variant. Then, several descriptors can be extracted from the matrix, e.g. the entropy value, whose volume is also in accordance to the regularity of the pattern; these descriptors are also known as Haralick features. However, we can face the situation, when we do not have a priori knowledge regarding the distance and direction of the possibly repeating texture elements. In these scenarios, the short position vectors in the major directions might lead to a less characteristic descriptor extracted from the co-occurrence matrix. The simplest discovery of the vectors revealing the repeating components could be based on an exhaustive (brute force) search with checking all the possible position vectors. However, this operation is really expensive especially when large images are to be analyzed.

To help with finding the appropriate position vectors, we introduce a methodology to recommend easily and in a short time a suitable vector to generate the co-occurrence matrix. Our approach is based on the Lenstra-Lenstra-Lovász (LLL) algorithm [56] with making it capable to test the

regularity of the pattern with automatically fitting a grid on a point cloud extracted from the pattern. The LLL algorithm is a polynomial-time one and was published in 1982. Since then, it has been successfully applied in several fields [28]. The algorithm can be applied very effectively for grid approximation for a given point set [27]. Using this technique, we can recommend position vectors spanning well-approximating grids for the creation of the co-occurrence matrix. We will also demonstrate that the extracted Haralick features using these vectors have similar characterizing performance than the ones calculated by vectors found via a brute-force search. In other words, our main contribution is to introduce an efficient method to recommend position vectors to extract secondary statistical descriptors.

The results presented in this section are published in my following publication: [81].

4.1.2 Finding well-approximating grids

The application of the LLL Algorithm

The LLL algorithm needs a set of points on which the grid will be fit. Therefore, as a first step we have to extract the reference points from the image we want to process. To do so, first we reduce the number of intensity values by simple quantization, then, we extract the centroids of larger components to act as the reference points. In this way, we can focus on the main repeats within the image with ignoring smaller intensity noises together with keeping the number of reference points relatively low.

Obtaining reference points via quantization

During quantization, the spatial resolution of the image is not changed, however, the number of intensity or color values can be drastically re-

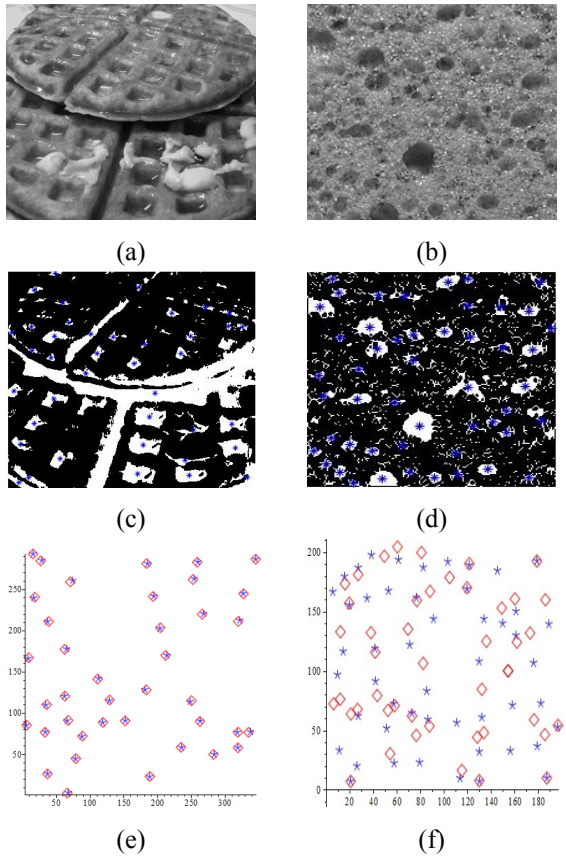


Figure 4.1: Finding the position vector by the LLL algorithm; a) regular image, b) irregular image, c)-d) quantized regular/irregular images with reference points, e)-f) best-approximating grids found by LLL.

duced. To perform quantization, the intensity range is split into equal parts or based on some kind of statistics. In our case, we consider grayscale images (see Figure 4.1(a)(b)), and the intensity range is divided into 7 parts, that is to say the image is quantized into seven levels. Naturally, the number of levels can be freely adjusted.

After quantization, we consider all the 7 intensity level images separately, and extract the centroids of the components that are present in these images as reference points (see Figure 4.1(c)(d)). That is, we run the LLL

algorithm on each of the 7 levels for the point sets. From these 7 images we keep that one for which the best approximating grid has been found (see Figure 4.1(e)(f)). In other words, as for regularity, we try to extract the most prominent repeating pattern.

The application of the LLL Algorithm

A grid itself is a set of integer linear combinations of some independent vectors. These vectors are also referred as the basis of the lattice. The LLL algorithm in its original form is a polynomial-time algorithm, which is able to perform basis reduction with providing a short vector in a new basis [56].

This algorithm becomes popular in several fields, mostly because of its polynomial running time and theoretical guarantees. With an appropriate modification of the original idea, an algorithm has been provided in [28] to find well-approximating grids on a set of points. From the error of the approximation one can also conclude to the regularity of the pattern, and the basis vectors spanning the approximating grids suggest some kind of repeat in the pattern. Note that in [28] the type (hexagon grid) and the error of the approximating grid are udes to decide, if the pigment network is typical or atypical. However this thesis defines regularity/irregularity of the pattern with the help of features deriving from the co-occurrence matrix calculated from the approximating grid.

A natural necessary limitation to apply this approach for pattern regularity analysis is to determine the expected approximation error for a random point set as a threshold. Namely, if the error is above this threshold, then the pattern cannot be considered to be regular. In other words, in such cases we do not use any vectors for extracting Haralick features via an occurrence matrix, since the pattern is considered to be irregular. In this thesis, we give an estimation for this threshold. Namely, to estimate

this figure, we have generated 1000 uniformly distributed random sets of points and tried to fit a grid on them with the help of the LLL algorithm. In each case, we have determined the approximation error of the best-approximating grid. Then, we have estimated the expected value and variance of the error distribution we have gained (see also Figure 4.2). The expected value of the error has been found to be 3.61, while the standard deviation 0.2457. Notice that, we did not have to perform any normalization regarding the number and volume of the random point sets, since the error term given in [28] includes that.

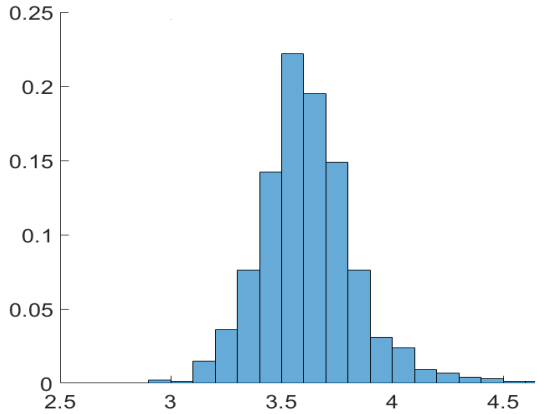


Figure 4.2: Probability distribution of the error of best-approximating grids using 1000 random sets of points.

4.1.3 Extracting position vectors to compose co-occurrence matrices.

In its (i, j) entry, the gray level co-occurrence matrix (GLCM or shortly $g(i, j)$) gives the number pixel-pairs having intensities i and j residing at a given distance $d \geq 1$ and direction $\theta \in [0, 2\pi)$. Thus, the co-occurrence matrices carry information about the intensity-variations depending on the distance and direction. Thus, for intensity levels i and j the GLCM is

calculated as

$$g(i,j) = \sum_{x=1}^{n} \sum_{y=1}^{n} (I(x,y) = i) \wedge (I(x',y') = j),$$

where the size of the original image I is $n \times n$, and

$$d = max\{|x - x'|, |y - y'|\}.$$

Notice that, in our terminology the position vector is defined by the pair (d, θ).

Several global measures can be derived from GLCMs, which are also known as Haralick features [20]. The most popular descriptors are:

1. $Entropy = \sum_i \sum_j g(i,j) \log g(i,j)$,
2. $Contrast = \sum_i \sum_j (i-j)^2 g(i,j)$,
3. $Energy = \sum_i \sum_j g(i,j)^2$,
4. $Variance = \sum_i \sum_j (i-\mu)^2 g(i,j)$,
5. $Homogeneity = \sum_i \sum_j \frac{1}{1+(i-j)^2} g(i,j)$,

where μ is the mean of $g(i,j)$ [63, 20].

To create the matrix we need a position vector. Usually, direct neighboring pixels are considered as basis to generate the matrix, i.e. the position vector has length $d = 1$, while the direction can be $\theta = k \cdot 45°$ with $k = 0, \ldots, 7$. In some applications longer vectors (e.g. for $d \in [2, \ldots, 10]$) are considered, however, no explicit recommendations are given for the proper selection [20]. For example, in regularity analysis the Haralick features are efficient, if they are calculated according to some repeats in the pattern (e.g. entropy gets smaller for more regular patterns). Thus,

for regularity analysis purposes, our objective is to find position vectors that fit on repeating components. Such vectors are also expected e.g. to provide small entropy values for regular patterns.

A cardinal issue regarding our approach is, whether the extraction of a position vector can be found more efficiently by the LLL algorithm than e.g. a brute-force search. As for brute-force, trying out all possible position vectors is a very expensive procedure since it has $O(n^3)$ complexity considering an image of size $n \times n$. In our experiments the sample images sizes varies from 128 to 400. The running time of the LLL algorithm recommended by us is much smaller, since its complexity is $O(p^4 \log d)$, where p is the number of the reference points, and d is the largest length of the grid basis vector under the Euclidean norm [56]. This calculation was done dedicated to our approach presented in this work. In this formula, the number of the reference points can be limited. In our experiments, we have found that $p = 50$ was a sufficient setting for an adequate grid approximation.

That is, the determination of the position vector with the current methodology is relatively quick. In the next section, we will demonstrate with empirical tests that our LLL-based approach is able to substitute the brute-force one, Namely, the position vectors found by its help lead to similar Haralick features. Especially, these features provides the smallest entropy figures, since the position vectors could reveal the repeats in the pattern.

4.1.4 Experimental results

To test our method we have collected 80 regular and 80 irregular images into an image set. As a comparative study, we have composed the co-occurrence matrices using the commonly used neighboring vectors, the vector proposed by the LLL algorithm, and also the vectors found by a brute force search. The last case meant that we have generated all possible

lengths and directions and co-occurrence matrices are calculated based on each vector. Then, the vector providing minimal entropy has been selected for each image.

As a well-known fact, the values of the Haralick descriptors (e.g. entropy, contrast and variance) are smaller/higher, when they are extracted from a GLCM derived from a regular/irregular pattern. So, after we produced all GLCMs using all possible vectors, we have selected that matrix which provided the smallest values regarding the extracted specific Haralick features in each cases. Notice that, the complexity of this step has been discussed for the LLL and brute force cases in previous section.

After the calculations of the GLCMs according to the three investigated cases, we have calculated all the mentioned Haralick descriptors for each image from our dataset and collected them to three different feature vectors. Then, we have classified the images as regular and irregular ones by a Naive Bayes [41], a Bayes Net [19] and a multi-layer perceptron classifier (MLPC) with 5 hidden layers [75]. We used 30% of the dataset for testing. As it can be seen in Table 4.1, the accuracy of the different classifiers varied in the three cases. The lowest accuracy is reached by the commonly used neighboring vectors, while the highest accuracy was achieved in the brute force case. However, it can be nicely observed that our LLL approach has a very similar performance to that of the brute force one suggesting a reliable substitutability. As a possible further improvement, we have completed this investigation with passing also the error of the best-approximating grid as an additional feature to the classifiers. With this extension the classification results belonging to the LLL case are slightly improved and got indeed very close to the brute force one (see the last row of Table 4.1).

Table 4.1: Classification results of regular and irregular patterns using different classifiers.

Method	Naive Bayes	Bayes Net	MLPC
neighboring vector	90%	90%	87.5%
brute force vector	97.81%	97.78%	96.2%
LLL vector	97.5%	97.5%	95%
LLL vector (plus error)	97.53%	97.63%	96.2%

4.1.5 Discussion and conclusions

The Haralick features derived from GLCMs based on different position vectors can be expected to be useful to determine whether medical texture patterns are regular or irregular. For example, this feature of GLCMs makes possible to analyze the regularity of polyps on endoscopic slides e.g. in peristole examinations. In this scenario, via the examination of capillary changes neoplastic lesions can be detected, and these irregular mutations of the capillary networks are the indicators of tumorous diseases. Our regularity measurement method can be used to categorize the capillary patterns into four different types according to their qualities. They can be meshed, surrounded by mucosal glands or non-uniform and nearly avascular (for more details see [84]) as it can be observed in Figure 4.3, as well.

To summarize, we have shown that the descriptors extracted from GLCMs can be used to determine whether a texture pattern is regular or irregular if the GLCMs are composed from longer position vectors. To take advantage of the information that the texture may contain repeating patterns, we try to find the appropriate distance and direction between two repeats. To do so, we extract reference points and look for well-approximating grids by the LLL algorithm. The vectors spanning the best-

fitting grids are used to compose GLCMs. According to our experimental results, our LLL-based approach to find position vectors is a valid substitution of the brute force search, whose complexity makes it an unsuitable solution. Moreover, we have also found that pattern classifiers might achieve higher accuracies, when they use feature vectors extracted by the help of position vectors longer than the commonly used neighboring ones.

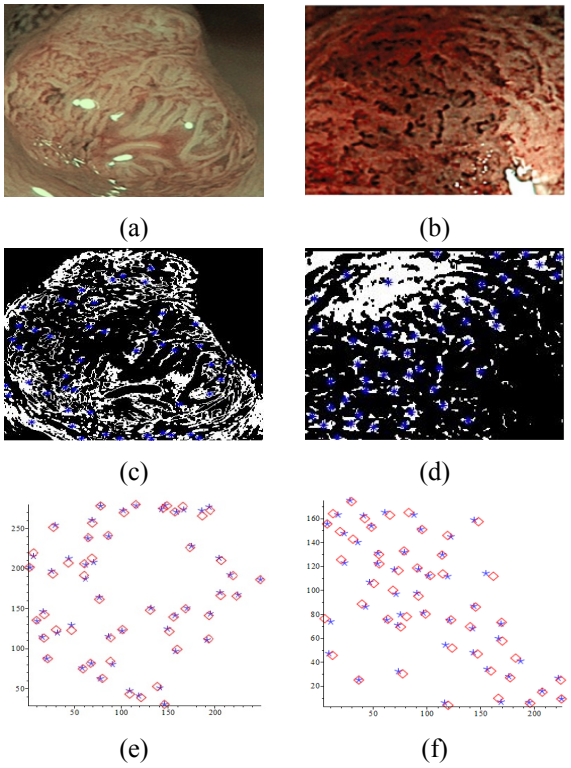


Figure 4.3: Capillary patten analysis; a) regular pattern, b) irregular pattern, c)-d) quantized regular/irregular image, e)-f) best-approximating grids.

4.2 Detecting Periodicity in Digital Images by the LLL Algorithm

4.2.1 Introduction

Patter analysis is a traditional task in digital image processing. In various fields the regularity/irregularity of the pattern directly relates to the underlying problem, so a proper decision on this phenomenon is the essence of the solution. To address the recognition of pattern regularity primarily some kind of periodicity check can be performed based on e.g. autocorrelation like in [49]. In [56] the authors have proposed a procedure based on the LLL algorithm to find best approximating grids to an input point set. The theory is worked out for point sets, which can be composed e.g. via the extraction of dominant digital image components and the representation of them with single points e.g. in terms of their centroids. However, segmentation errors may occur during this extraction steps causing holes in the pattern. Such a scenario can be observed in Figure 4.4, where our intention is to extract pigment networks from skin lesion images and classify them as typical (regular pattern) or atypical (irregular one). In [28], it was only required that an approximating grid point should fall in a close environment of each point in the input set. However, this error measurement ignores possible holes in the input pattern since does not punish the reversed cases, when there are no base points close to the approximating grid points. Thus, to resolve this issue now we complete error measurement with a complementary check that the number of the approximating grid points should be close to that of the cardinality of the input point set. The proper extra condition can be formulated by counting lattice points in the convex hull of the original point set.

The results presented in this section are published in my following publication: [31].

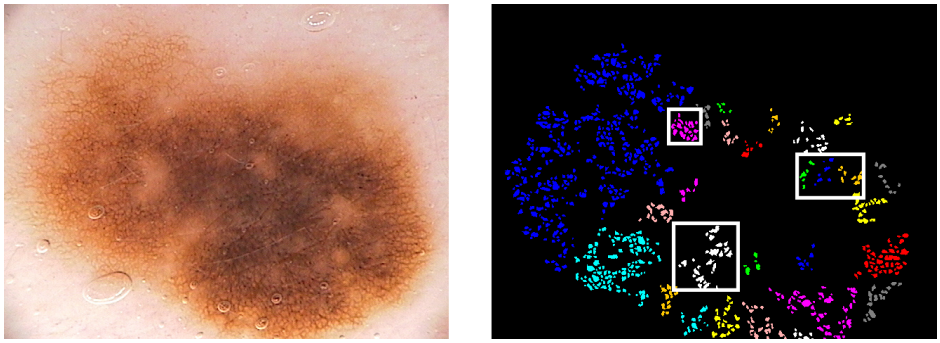


Figure 4.4: Segmentation results for a skin lesion image to extract pigment network; original image (left) and some false segmentation results (right) causing holes in the pattern are marked with white rectangle.

4.2.2 Periodicity and lattices

Suppose that we observe the occurrence of a certain pattern on a digital image repeatedly, and we wonder whether this occurrence can be regarded as periodic, or not. Such questions appear in several problems of image processing.

As a first step in building our model, we assume that we have to deal with a finite number of points on the plain. This can be achieved by standard discretization techniques, e.g. after considering the centroids of the occurring copies of the pattern.

Now let H be a subset of $\mathbb{R}^2$. Following the standard terminology, we say that H is periodic if there exist linearly independent vectors $\underline{u}, \underline{v} \in \mathbb{R}^2$ such that for any $\underline{h}$ in H, the vectors $\underline{h} \pm \underline{u}$ and $\underline{h} \pm \underline{v}$ are in H as well. Let

$$\Lambda = \{x\underline{u} + y\underline{v} : x, y \in \mathbb{Z}\}$$

be the lattice generated by $\underline{u}, \underline{v}$ in $\mathbb{R}^2$. Then the periodicity of H implies that $H = H + \Lambda$. (Indeed, $H \subseteq H + \Lambda$ is obvious, while $H + \Lambda \subseteq H$ follows directly from the definition, by noting that $h + xu + yv$ $(x, y \in \mathbb{Z})$ can be obtained from h by adding $|x|$ times u or $-u$, and $|y|$ times v or $-v$.)

Clearly, any periodic set in $\mathbb{R}^2$ has to be infinite. However, we obviously need to consider the periodic property of *finite* subsets of $\mathbb{R}^2$. For this, first observe that if H is a countably infinite subset of $\mathbb{R}^2$ and H is periodic, then H is a shifted lattice, that is, H is of the shape $H = \underline{o} + \Lambda =: \Lambda'$ for some $\underline{o} \in \mathbb{R}^2$ and lattice Λ in $\mathbb{R}^2$. Let now T be a finite subset of Λ' with Λ' as above. Adopting the discrete convexity notion of Kim [43] (see also [44]), we say that T is convex if $T^c \cap \Lambda' = T$, where T^c is the convex hull of T in $\mathbb{R}^2$. Altogether, this is the property we use for the definition of periodicity. That is, a finite subset T of $\mathbb{R}^2$ is called periodic if there exists a shifted lattice Λ' in $\mathbb{R}^2$ such that T is a convex subset of Λ'. To measure the periodicity of finite subsets T of shifted lattices, we introduce the following function:

$$\text{per}(T) = \min_{\Lambda'} \frac{|T|}{|T^c \cap \Lambda'|},$$

where $|S|$ denotes the number of elements of a set S and the minimum is taken over all shifted lattices Λ' containing T. (Clearly, this minimum exists.) In this way, T is periodic if and only if $\text{per}(T) = 1$. (Indeed, it is clear that if T is periodic then $\text{per}(T) = 1$. On the other hand, if $\text{per}(T) = 1$ then there exists a Λ' as above, containing T, such that $|T| = |T^c \cap \Lambda'|$. As $T \subset T^c \cap \Lambda'$, the equality $|T| = |T^c \cap \Lambda'|$ shows that in fact $T = T^c \cap \Lambda'$, so T is a convex subset of Λ'.)

Then we can measure (decide about) the periodicity of an arbitrary finite subset $S = \{s_1, \ldots, s_k\} \subset \mathbb{R}^2$ of cardinality k in the following way.

Step 1. Following the method of Hajdu, Hajdu and Tijdeman [27] (based upon the LLL algorithm, see [56]) we can find a 'well approximating' shifted lattice Λ' for S. Namely, let the error of the approximation be calculated as

$$E_{approx} := \frac{\sqrt{\sum_{s \in S} |s - \Lambda'|^2}}{\Delta} \left(\frac{\text{diam } S}{\Delta} \right)^{\frac{2}{k-3}},$$

where diam S is the diameter of the point set S, and Δ is the square root of the lattice determinant of Λ'. If E_{approx} is 'too large' (see the paper of Tiba, Harangi and A. Hajdu [81] for experiments; c.f. also [27] and [28]), then we can immediately say that the pattern in question does not appear periodically, and the forthcoming steps are superfluous.

Step 2. Write S' for the points of $\Lambda' = \underline{o} + \Lambda$ (with the previous notation) corresponding to the (approximated) points of S, and let $\underline{u}, \underline{v}$ be a basis of Λ. (They can be obtained by the already mentioned method of Hajdu, Hajdu and Tijdeman [27].) Put

$$A := \{(x, y) \in \mathbb{Z}^2 : \underline{o} + x\underline{u} + y\underline{v} \in S'\}.$$

Observe that $\mathrm{per}(S') = \mathrm{per}(A)$.

Step 3. Find a sublattice L of $\mathbb{Z}^2$ of largest index containing A. (Typically, L will be $\mathbb{Z}^2$ itself.) For this, observe that $L = \sum_{(x,y)\in A} (x, y)\mathbb{Z}$. Thus it is standard to find a basis $\underline{p}, \underline{q}$ of L; see e.g. p. 73 of Cohen's book [14], where an algorithm based upon the Hermite normal form of integer matrices is given. Then, transform A as follows:

$$B := \{(x, y) \in \mathbb{Z}^2 : x\underline{p} + y\underline{q} \in A\}.$$

In this way we have

$$\mathrm{per}(S') = \mathrm{per}(B) = \frac{|B|}{|B^c \cap \mathbb{Z}^2|}.$$

Step 4. Note that $|B| = |S'|$, so this number can be calculated easily. The number $|B^c \cap \mathbb{Z}^2|$ can be obtained in a very efficient way, based upon Barvinok's algorithm [7]. We use the Maple 15 [8] implementation of Baldoni, Berline and Vergne [6]. So, altogether we have an efficient way to calculate $\mathrm{per}(S')$.

Step 5. We can combine the error of approximation E_{approx} obtained in Step 1 and the measure $\mathrm{per}(S')$, e.g. say using a threshold, for deciding about the periodicity of S.

We illustrate our method by a simple example.

Example 1. Let our starting set of points be given by

$$S = \{(0,0), (3.218875824, 3.891820298), (4.007333185, 4.510859506),$$

$$(4.795790546, 5.129898714), (6.405228458, 7.075808863),$$

$$(8.014666370, 9.021719012)\}.$$

In Step 1, we get $S' = S$ together with $\underline{u} = (1.609437912, 1.945910149)$, $\underline{v} = (4.007333185, 4.510859506)$. Further, we get $E_{approx} = 0$.

In Step 2, we obtain

$$A = \{(0,0), (0,-2), (-1,0), (-2,2), (-2,1), (-2,0)\},$$

where the elements in A are the coefficients of the elements of S' in the basis $\underline{u}, \underline{v}$, in the given order. (In this case we have $\underline{o} = (0,0)$.)

In Step 3, we see that $L = \mathbb{Z}^2$ and $\underline{p} = (1,0)$, $\underline{q} = (0,1)$. Thus

$$B = A = \{(0,0), (0,-2), (-1,0), (-2,2), (-2,1), (-2,0)\}.$$

This follows from the fact that the gcd of the 2×2 subdeterminants of the matrix

$$\begin{pmatrix} 0 & 0 & -1 & -2 & -2 & -2 \\ 0 & -2 & 0 & 2 & 1 & 0 \end{pmatrix}$$

(composed of the entries of the elements of A) is 1.

In Step 4, using the Maple code of Baldoni, Berline and Vergne we obtain that $|B^c| = 9$. (In fact, in this simple case B^c is a 2×2 square, with vertices $(-2,0), (0,-2), (0,0), (-2,2)$.) Hence we get

$$\text{per}(S') = \frac{|B|}{|B^c|} = \frac{6}{9} = \frac{2}{3}.$$

In Step 5, based upon $E_{approx} = 0$ and $\text{per}(S') = 2/3$, depending on the actual application we are dealing with, we can decide whether we consider S to be periodic or not.

4.2.3 Application to pigment network segmentation

As we have presented in the introduction, checking the periodicity (gridness) of a point set was motivated by the regularity analysis of pigment networks in skin lesion images. In this task we exploit the method introduced in the previous sections to check whether the extraction of the components on a pigment network was successful or not. The latter case generally occurs when our detector algorithm misses some components causing holes in the extracted pattern. With the proper details are given in [28], the extraction of the pigment cells can be summarized as follows:

- the input color image is converted to grayscale,
- for each pixel, the intensity profiles of lines passing through the given pixel are considered,
- second order derivative of the Gaussian filters are matched to the profiles,
- large filter response values are considered for pigment hole candidates,
- a hysteresis thresholding technique is applied to this response map for the final network components.

As it can be seen above, the number of extracted components can be increased with a corresponding threshold. Consequently, when a low periodicity score is found for an extracted pigment network, we lower the threshold to eliminate some holes in the pattern. As a demonstrative example, Figure 4.5 depicts such a scenario, when an extracted network with a low periodicity score (0.14) has been improved to a higher score (0.91), since re-segmentation with a lower threshold has found more network components. As simple technical issues note that the point set is generated as the centroids of the binary components, and using low threshold

in our segmentation method in an unjustified way is risky, since it can lead to over-segmentation.

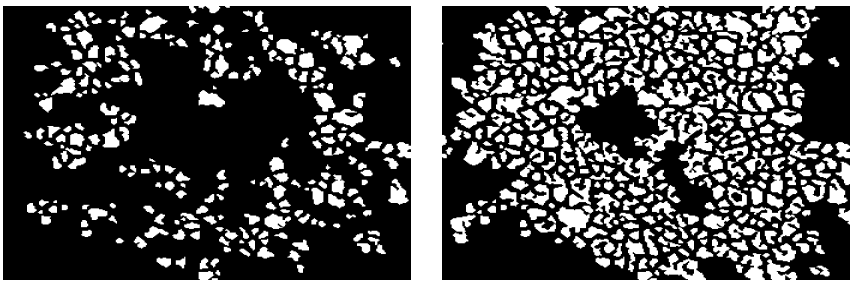


Figure 4.5: Segmented pigment network in a skin lesion image with a low periodicity score (left) and the result of re-segmentation with a higher periodicity score (right).

Acknowledgements

Some element of this thesis was supported by the ÚNKP-20-4-I New National Excellence Program of the Ministry for Innovation and Technology from the Source of the National Research, Development and Innovation Fund.

I am grateful to my thesis advisor András Hajdu for his encouragement and guidance. I am also thankful to the Doctoral School of Informatics of the University of Debrecen for having provided the opportunity to conduct researches and to write my thesis.

Finally, I would like to thank God and my family for their support and patience throughout the years.

Summary

In this section we summarize the contents of the dissertation in thesis points.

My results presented in chapter 2 are summarised in the following thesis points:

TP 1.1 I have created a novel stochastic model for ensemble-based binary classification that considers the energy as the joint probability function of the member accuracies. I have theoretically proved and also validated experimentally that it can be efficiently incorporated in a stochastic search process as a stopping rule to find accurate ensembles.

TP 1.2 I have provided a novel and efficient stochastic search method that better fits the given energy, which can be incorporated in other stochastic strategies as well. I have showed the dominance of the method over other state-of-the-art ones using publicly available databases and standard metrics.

TP 1.3 I have derived new theoretical results regarding the expected ensemble accuracy and its variance for the multiclass classification problem under a Knapsack constraint. I have verified the results in the optic disc detection problem in retinal images.

Corresponding publications: [1], [5].

In chapter 3, we show applications of ensemble-based methods in several areas.

First, results for the COVID-19 pandemic are presented, for which I summarise my own results below:

TP 2.1 I have developed a novel method to predict the new cases of COVID-19 using an ensemble of Recurrent Neural Networks, adding extra features, and combining transfer learning with a complex architecture of interconnected subnetworks. I have tested the approach on official data from the first and second waves of COVID-19 and compared it with other state-of-the-art methods. The new model provides a more realistic prediction over a longer period.

Corresponding publication: [2].

In the last part of chapter 3, we present our results on outlier detection, also using an ensemble approach.

There are numerous reasons why inappropriate data can occur in a database. It is essential to detect and eliminate these elements for getting accurate results and conclusions. The process of filtering anomalies generated in the database is called outlier detection. The outliers, that are extreme values deviating from other observations on data, indicate a variability in a measurement, experimental errors or a novelty.

The thesis point below summarizes my related findings.

TP 2.2 I have created an ensemble-based outlier detection method, where the members of the ensemble are Convolutional Neural Networks (CNNs) and a Support Vector Machine (SVM) is used as a classifier. I considered majority voting for aggregation which results in a very accurate outlier filtering. I have

evaluated the performance of the proposed method for filtering databases consisting of retinal and skin lesion images, respectively. The results show that the proposed ensemble system improved the effectiveness of the member-level components in the outlier detection for both retinal and skin lesion images.

Corresponding publication: [3].

In chapter 4, our optimization approaches for measuring the regularity of image patterns are presented.

Co-occurrence matrices as sources of second order statistical descriptors are commonly used in texture classification tasks. To generate such a matrix, we need a position vector to check possible intensity frequencies in its endpoints. My corresponding results are enclosed in the following two thesis points.

TP 3.1 I have composed an novel algorithm on a theoretical basis to locate such position vectors according which the pattern of a texture repeats and thus, the descriptors (Haralick features) derived from the co-occurrence matrix are capable to characterize the regularity of it. For this purpose I have determined well-approximating grids by the LLL algorithm, which has a polynomial running time providing a much more efficient solution than brute force search.

TP 3.2 I have evaluated the method on an own image set. I have composed the co-occurrence matrices using the commonly used position vectors, the vector proposed by the LLL algorithm, and also the vectors found by brute force search. I have classified the images as regular and irregular ones by a Naive Bayes classifier, a Bayes Net and a multi-layer perceptron classifier

(MLPC) with 5 hidden layers. The proposed approach has given the closest solutions to the optimal (brute force-based) ones.

Corresponding publication: [6].

TP 3.3 I have provided a new algorithm to decide whether some repeatedly occurring pattern in a digital image has periodical nature. Accordingly, I have extracted textural elements and represented them by single pixels. I have used lattice theory and the LLL algorithm to fit lattices to this point set, and an efficient counting method of Barvinok to determine regularity. After some appropriate transformations I have considered the convex hull to detect and punish lattice points fitted on missing pattern components.

Corresponding publication: [4].

UNIVERSITY AND NATIONAL LIBRARY
UNIVERSITY OF DEBRECEN
H-4002 Egyetem tér 1, Debrecen
Phone: +3652/410-443, email: publikaciok@lib.unideb.hu

Registry number: DEENK/358/2022.PL
Subject: PhD Publication List

Candidate: Attila Tiba
Doctoral School: Doctoral School of Informatics
MTMT ID: 10068658

List of publications related to the dissertation

Foreign language scientific articles in international journals (2)

1. Hajdu, A., Terdik, G., **Tiba, A.**, Tomán, H.: A stochastic approach to handle resource constraints as knapsack problems in ensemble pruning.
Mach. Learn. 111, 1551-1595, 2022. ISSN: 0885-6125.
DOI: http://dx.doi.org/10.1007/s10994-021-06109-0
IF: 5.414 (2021)

2. Kolozsvári, L. R., Bérczes, T., Hajdu, A., Gesztelyi, R., **Tiba, A.**, Varga, I., Al-Tammemi, A. B., Szőllősi, G. J., Kolozsváriné Harsányi, S., Garbóczy, S., Zsuga, J.: Predicting the epidemic curve of the coronavirus (SARS-CoV-2) disease (COVID-19) using artificial intelligence: an application on the first and second waves.
Informatics in Medicine Unlocked. 25, 1-13, 2021. ISSN: 2352-9148.
DOI: http://dx.doi.org/10.1016/j.imu.2021.100691

Foreign language conference proceedings (4)

3. **Tiba, A.**, Bartik, Z., Tomán, H., Hajdu, A.: Detecting outlier and poor quality medical images with an ensemble-based deep learning system.
In: 11th International Symposium on Image and Signal Processing and Analysis (ISPA), Ieee-Inst Electrical Electronics Engineers Inc, Piscataway, 99-104, 2019. ISBN: 9781728131405

4. Hajdu, L., Harangi, B., **Tiba, A.**, Hajdu, A.: Detecting Periodicity in Digital Images by the LLL Algorithm.
In: Progress in Industrial Mathematics at ECMI 2018. Ed.: István Faragó, Ferenc Izsák, Péter L. Simon, Springer, Cham, 613-619, 2019, (Mathematics in Industry ; 30.)(The European Consortium for Mathematics in Industry ; 30.) ISBN: 9783030275495

5. **Tiba, A.**, Hajdu, A., Terdik, G., Tomán, H.: Optimizing Majority Voting Based Systems Under a Resource Constraint for Multiclass Problems.
In: Progress in Industrial Mathematics at ECMI 2018. Ed.: István Faragó, Ferenc Izsák, Péter L. Simon, Springer, Cham, 529-534, 2019, (Mathematics in Industry ; 30.)(The European Consortium for Mathematics in Industry ; 30.) ISBN: 9783030275495

6. **Tiba, A.**, Harangi, B., Hajdu, A.: Efficient Texture Regularity Estimation for Second Order Statistical Descriptors.
In: Proceedings of the 10th International Image and Signal Processing and Analysis (ISPA). Ed.: Stanislav Kovačič, Sven Lončarić, Matej Kristan, Vitomir Štruc, Mladen Vučić, University of Zagreb, Zagreb, 90-94, 2017. ISBN: 9781509040117

List of other publications

Hungarian scientific articles in Hungarian journals (2)

7. Bogacsovics, G., Hajdu, A., Harangi, B., Lakatos, I., Lakatos, R., Szabó, M., **Tiba, A.**, Tóth, J., Tarcsi, Á.: Adatelemzési folyamat és keretrendszer a közigazgatás számára.
Közigazgatástudomány. 1 (2), 146-158, 2021. ISSN: 2786-1910.
DOI: http://dx.doi.org/10.54200/kt.v1i2.24

8. Bogacsovics, G., Hajdu, A., Harangi, B., Lakatos, I., Lakatos, R., Szabó, M., **Tiba, A.**, Tóth, J.: Napelemfarmok Magyarország területén történő elhelyezését segítő döntéstámogató rendszer fejlesztése.
Közigazgatástudomány. 1 (2), 134-145, 2021. ISSN: 2786-1910.
DOI: http://dx.doi.org/10.54200/kt.v1i2.23

Foreign language scientific articles in Hungarian journals (3)

9. Lantang, O., Terdik, G., Hajdu, A., **Tiba, A.**: Comparison of single and ensemble-based convolutional neural networks for cancerous image classification.
Ann. Math. Inform. 54, 45-56, 2021. ISSN: 1787-5021.
DOI: http://dx.doi.org/10.33039/ami.2021.03.013

10. Lantang, O., Terdik, G., Hajdu, A., **Tiba, A.**: Investigation of the efficiency of an interconnected convolutional neural network by classifying medical images.
Ann. Math. Inform. 53, 219-234, 2021. ISSN: 1787-5021.
DOI: http://dx.doi.org/10.33039/ami.2021.04.001

11. Bogacsovics, G., Hajdu, A., Lakatos, R., Beregi-Kovács, M., **Tiba, A.**, Tomán, H.: Replacing the SIR epidemic model with a neural network and training it further to increase prediction accuracy.
Ann. Math. Inform. 53, 73-91, 2021. ISSN: 1787-5021.
DOI: http://dx.doi.org/10.33039/ami.2021.02.003

UNIVERSITY of DEBRECEN

UNIVERSITY AND NATIONAL LIBRARY
UNIVERSITY OF DEBRECEN
H-4002 Egyetem tér 1, Debrecen
Phone: +3652/410-443, email: publikaciok@lib.unideb.hu

Foreign language scientific articles in international journals (1)

12. Bankó, C., Nagy, Z. L., Nagy, M., Szemán-Nagy, G., Rebenku, I., Imre, L., **Tiba, A.**, Hajdu, A., Szöllősi, J., Kéki, S., Bacsó, Z.: Isocyanide Substitution in Acridine Orange Shifts DNA Damage-Mediated Phototoxicity to Permeabilization of the Lysosomal Membrane in Cancer Cells.
Cancers (Basel). 13 (22), 1-24, 2021. EISSN: 2072-6694.
DOI: http://dx.doi.org/10.3390/cancers13225652
IF: 6.575

Foreign language conference proceedings (2)

13. Lantang, O., **Tiba, A.**, Hajdu, A., Terdik, G.: Convolutional Neural Network For Predicting The Spread of Cancer.
In: Proceedings of the 10th IEEE International Conference on Cognitive Infocommunications : CogInfoCom 2019. Szerk.: Péter Baranyi, IEEE-Inst Electrical Electronics Engineers Inc, Piscataway, 175-180, 2019. ISBN: 9781728147932

14. Bérczes, A., Bérczes, T., Varga, I., **Tiba, A.**, Zsuga, J.: Using Laplacian spectrum to analise the comorbidities network of hemorrhagic stroke.
In: Proceedings of the 10th IEEE International Conference on Cognitive Infocommunications : CogInfoCom 2019. Szerk.: Péter Baranyi, IEEE-Inst Electrical Electronics Engineers Inc, Piscataway, 53-60, 2019. ISBN: 9781728147932

Total IF of journals (all publications): 11,989
Total IF of journals (publications related to the dissertation): 5,414

The Candidate's publication data submitted to the iDEa Tudóstér have been validated by DEENK on the basis of the Journal Citation Report (Impact Factor) database.

22 September, 2022

Összefoglaló

Ebben a részben tézispontokban foglaljuk össze a disszertáció tartalmát.

A második fejezetben bemutatott eredményeket a következő tézispontokban összegezzük:

TP 1.1 Egy új sztochasztikus modellt hoztam létre az együttes-alapú bináris osztályozáshoz, amely az energiát a tagok pontosságának együttes valószínűségi függvényeként tekinti. Elméleti úton bebizonyítottam és kísérletileg igazoltam, hogy ez a modell hatékonyan beépíthető egy sztochasztikus keresési folyamatba, mint megállási szabály a pontos együttes rendszerek megtalálása érdekében.

TP 1.2 Egy újszerű és hatékony, az energiához jobban illeszkedő sztochasztikus keresési módszert adtam meg, amely más sztochasztikus stratégiákba is beépíthető. Nyilvánosan elérhető adatbázisok és szabványos metrikák segítségével kimutattam a módszer hatékonyságát más korszerű módszerekkel szemben.

TP 1.3 Új elméleti eredményeket vezettem le a várható együttes pontosság és annak varianciájára vonatkozóan többosztályos osztályozási feladatra a hátizsák probléma kényszerfeltétele mellett. Az eredményeket egy speciális objektumdetektálási problémán értékeltem ki, amely célja a vakfolt megtalálása retinális képeken.

Kapcsolódó publikációk: [1], [5].

A harmadik fejezetben az együttes-alapú módszerek alkalmazását mutatjuk be néhány területen. Először a COVID-19 világjárványra vonatkozó eredmények kerülnek bemutatásra, amelyekre vonatkozóan az alábbiakban foglaljuk össze az eredményeket:

TP 2.1 Új módszert dolgoztam ki a COVID-19 új eseteinek előrejelzésére rekurrens neurális hálózatok együttesének felhasználásával, extra jellemzők hozzáadásával, és a transzfer tanulást kombinálva az összekapcsolt alhálózatok komplex architektúrájával. A megközelítést a COVID-19 világjárvány első és második hullámának adatain teszteltem, és összehasonlítottam más korszerű módszerekkel. Az új modell reálisabb előrejelzést biztosít hosszabb időszakra vonatkozóan.

Kapcsolódó publikáció: [2].

A harmadik fejezet utolsó részében bemutatjuk a kiugró értékek észlelésével kapcsolatos eredményeinket, szintén egy együttes megközelítést alkalmazva.

Számos oka lehet annak, hogy egy adatbázisban nem megfelelő adatok fordulnak elő. A pontos eredmények és következtetések levonásához elengedhetetlen ezen elemek felderítése és kiküszöbölése. Az adatbázisban keletkező anomáliák kiszűrésének folyamatát nevezzük kiugró érték detektálásának. A kiugró értékek, amelyek az adatok többi megfigyeléseitől eltérő szélsőséges értékek, egy mérés változékonyságára, kísérleti hibákra vagy újdonságra utalnak.

Az alábbi tézispont összefoglalja az ezzel kapcsolatos megállapításainkat:

TP 2.2 Létrehoztam egy együttes-alapú kiugró érték detektálási módszert, ahol az együttes tagjai konvolúciós neurális hálózatok

(CNN) egy SVM (Support Vector Machine) osztályozóval kombinálva. Az aggregáláshoz többségi szavazást használtam, ami nagyon pontos kiugró érték szűrést eredményezett. A javasolt módszer teljesítményét retina-, illetve bőrelváltozásokat tartalmazó képekből álló adatbázisok szűrésére kiértékeltem. Az eredmények azt mutatják, hogy a javasolt együttes rendszer javította az egyedi tagok hatékonyságát a kiugró értékek felismerésében mind a retinális, mind a bőrelváltozásos képek esetében.

Kapcsolódó publikáció: [3].

A megyedik fejezetben a képminták szabályosságának mérésére szolgáló optimalizálási megközelítéseink kerülnek bemutatásra. Az együttes előfordulási mátrixokat mint másodrendű statisztikai leírók forrásait gyakran használják a textúra osztályozási feladatokban. Egy ilyen mátrix létrehozásához szükségünk van egy helyzetvektorra, amelynek végpontjaiban ellenőrizni tudjuk a lehetséges intenzitás értékeket.

Az ennek megfelelő új eredményeket a következő három tézispontba foglalja össze:

TP 3.1 Összeállítottam egy hatékony algoritmust olyan helyzetvektorok felkutatására, amelyek szerint a textúra mintázata ismétlődik, és így az együttes előfordulási mátrixból származtatott leírók (Haralick jellemzők) képesek jellemezni ezen minta szabályosságát. Erre a célra jól közelítő rácsokat határoztam meg az LLL algoritmus segítségével, amely polinomiális futási idővel rendelkezik, így sokkal hatékonyabb megoldást nyújt, mint a brute-force algoritmuson alapuló keresés.

TP 3.2 Az előző tézispontban meghatározott módszert egy saját képkészleten értékeltem ki. Az együttes előfordulási mátrixokat az általánosan használt szomszédsági vektorok, az LLL

algoritmus által javasolt vektor, valamint a brute-force eljárással végzett kereséssel talált vektorok segítségével állítottam össze. Ezután a képeket Naive Bayes, Bayes-háló és 5 rejtett réteggel rendelkező többrétegű perceptron osztályozó (MLPC) segítségével osztályoztam. A javasolt megközelítés az optimális (brute-force alapuló) megoldásokhoz nagyon közeli megoldásokat adott.

TP 3.3 Egy új algoritmust adtam meg annak eldöntésére, hogy egy digitális képen ismétlődően előforduló minta periodikusnak tekinthető-e vagy sem. Ennek érdekében kivontam a textúra elemeket, és egyetlen képponttal ábrázoltam őket. A rácselméletet és az LLL algoritmust használtam arra, hogy rácsokat illesszek erre a ponthalmazra, valamint Barvinok hatékony számlálási módszerét a szabályosság meghatározására. Néhány megfelelő transzformáció után a konvex burkot vettem figyelembe a hiányzó mintakomponensekre illesztett rácspontok felderítésére és büntetésére.

Kapcsolódó publikációk: [4], [6].

DEBRECENI EGYETEM
EGYETEMI ÉS NEMZETI KÖNYVTÁR
H-4002 Debrecen, Egyetem tér 1, Pf.: 400
Tel.: 52/410-443, e-mail: publikaciok@lib.unideb.hu

Nyilvántartási szám: DEENK/358/2022.PL
Tárgy: PhD Publikációs Lista

Jelölt: Tiba Attila
Doktori Iskola: Informatikai Tudományok Doktori Iskola
MTMT azonosító: 10068658

A PhD értekezés alapjául szolgáló közlemények

Idegen nyelvű tudományos közlemények külföldi folyóiratban (2)

1. Hajdu, A., Terdik, G., **Tiba, A.**, Tomán, H.: A stochastic approach to handle resource constraints as knapsack problems in ensemble pruning.
Mach. Learn. 111, 1551-1595, 2022. ISSN: 0885-6125.
DOI: http://dx.doi.org/10.1007/s10994-021-06109-0
IF: 5.414 (2021)

2. Kolozsvári, L. R., Bérczes, T., Hajdu, A., Gesztelyi, R., **Tiba, A.**, Varga, I., Al-Tammemi, A. B., Szőllősi, G. J., Kolozsváriné Harsányi, S., Garbóczy, S., Zsuga, J.: Predicting the epidemic curve of the coronavirus (SARS-CoV-2) disease (COVID-19) using artificial intelligence: an application on the first and second waves.
Informatics in Medicine Unlocked. 25, 1-13, 2021. ISSN: 2352-9148.
DOI: http://dx.doi.org/10.1016/j.imu.2021.100691

Idegen nyelvű konferencia közlemények (4)

3. **Tiba, A.**, Bartik, Z., Tomán, H., Hajdu, A.: Detecting outlier and poor quality medical images with an ensemble-based deep learning system.
In: 11th International Symposium on Image and Signal Processing and Analysis (ISPA), Ieee-Inst Electrical Electronics Engineers Inc, Piscataway, 99-104, 2019. ISBN: 9781728131405

4. Hajdu, L., Harangi, B., **Tiba, A.**, Hajdu, A.: Detecting Periodicity in Digital Images by the LLL Algorithm.
In: Progress in Industrial Mathematics at ECMI 2018. Ed.: István Faragó, Ferenc Izsák, Péter L. Simon, Springer, Cham, 613-619, 2019, (Mathematics in Industry ; 30.)(The European Consortium for Mathematics in Industry ; 30.) ISBN: 9783030275495

5. **Tiba, A.**, Hajdu, A., Terdik, G., Tomán, H.: Optimizing Majority Voting Based Systems Under a Resource Constraint for Multiclass Problems.
In: Progress in Industrial Mathematics at ECMI 2018. Ed.: István Faragó, Ferenc Izsák, Péter L. Simon, Springer, Cham, 529-534, 2019, (Mathematics in Industry ; 30.)(The European Consortium for Mathematics in Industry ; 30.) ISBN: 9783030275495

6. **Tiba, A.**, Harangi, B., Hajdu, A.: Efficient Texture Regularity Estimation for Second Order Statistical Descriptors.
In: Proceedings of the 10th International Image and Signal Processing and Analysis (ISPA). Ed.: Stanislav Kovačič, Sven Lončarić, Matej Kristan, Vitomir Štruc, Mladen Vučić, University of Zagreb, Zagreb, 90-94, 2017. ISBN: 9781509040117

További közlemények

Magyar nyelvű tudományos közlemények hazai folyóiratban (2)

7. Bogacsovics, G., Hajdu, A., Harangi, B., Lakatos, I., Lakatos, R., Szabó, M., **Tiba, A.**, Tóth, J., Tarcsi, Á.: Adatelemzési folyamat és keretrendszer a közigazgatás számára.
Közigazgatástudomány. 1 (2), 146-158, 2021. ISSN: 2786-1910.
DOI: http://dx.doi.org/10.54200/kt.v1i2.24

8. Bogacsovics, G., Hajdu, A., Harangi, B., Lakatos, I., Lakatos, R., Szabó, M., **Tiba, A.**, Tóth, J.: Napelemfarmok Magyarország területén történő elhelyezését segítő döntéstámogató rendszer fejlesztése.
Közigazgatástudomány. 1 (2), 134-145, 2021. ISSN: 2786-1910.
DOI: http://dx.doi.org/10.54200/kt.v1i2.23

Idegen nyelvű tudományos közlemények hazai folyóiratban (3)

9. Lantang, O., Terdik, G., Hajdu, A., **Tiba, A.**: Comparison of single and ensemble-based convolutional neural networks for cancerous image classification.
Ann. Math. Inform. 54, 45-56, 2021. ISSN: 1787-5021.
DOI: http://dx.doi.org/10.33039/ami.2021.03.013

10. Lantang, O., Terdik, G., Hajdu, A., **Tiba, A.**: Investigation of the efficiency of an interconnected convolutional neural network by classifying medical images.
Ann. Math. Inform. 53, 219-234, 2021. ISSN: 1787-5021.
DOI: http://dx.doi.org/10.33039/ami.2021.04.001

11. Bogacsovics, G., Hajdu, A., Lakatos, R., Beregi-Kovács, M., **Tiba, A.**, Tomán, H.: Replacing the SIR epidemic model with a neural network and training it further to increase prediction accuracy.
Ann. Math. Inform. 53, 73-91, 2021. ISSN: 1787-5021.
DOI: http://dx.doi.org/10.33039/ami.2021.02.003

DEBRECENI EGYETEM
EGYETEMI ÉS NEMZETI KÖNYVTÁR
H-4002 Debrecen, Egyetem tér 1, Pf.: 400
Tel.: 52/410-443, e-mail: publikaciok@lib.unideb.hu

Idegen nyelvű tudományos közlemények külföldi folyóiratban (1)

12. Bankó, C., Nagy, Z. L., Nagy, M., Szemán-Nagy, G., Rebenku, I., Imre, L., **Tiba, A.**, Hajdu, A., Szöllősi, J., Kéki, S., Bacsó, Z.: Isocyanide Substitution in Acridine Orange Shifts DNA Damage-Mediated Phototoxicity to Permeabilization of the Lysosomal Membrane in Cancer Cells.
Cancers (Basel). 13 (22), 1-24, 2021. EISSN: 2072-6694.
DOI: http://dx.doi.org/10.3390/cancers13225652
IF: 6.575

Idegen nyelvű konferencia közlemények (2)

13. Lantang, O., **Tiba, A.**, Hajdu, A., Terdik, G.: Convolutional Neural Network For Predicting The Spread of Cancer.
In: Proceedings of the 10th IEEE International Conference on Cognitive Infocommunications : CogInfoCom 2019. Szerk.: Péter Baranyi, IEEE-Inst Electrical Electronics Engineers Inc, Piscataway, 175-180, 2019. ISBN: 9781728147932

14. Bérczes, A., Bérczes, T., Varga, I., **Tiba, A.**, Zsuga, J.: Using Laplacian spectrum to analise the comorbidities network of hemorrhagic stroke.
In: Proceedings of the 10th IEEE International Conference on Cognitive Infocommunications : CogInfoCom 2019. Szerk.: Péter Baranyi, IEEE-Inst Electrical Electronics Engineers Inc, Piscataway, 53-60, 2019. ISBN: 9781728147932

A közlő folyóiratok összesített impakt faktora: 11,989
A közlő folyóiratok összesített impakt faktora (az értekezés alapjául szolgáló közleményekre): 5,414

A DEENK a Jelölt által az iDEa Tudóstérbe feltöltött adatok bibliográfiai és tudománymetriai ellenőrzését a tudományos adatbázisok és a Journal Citation Reports Impact Factor lista alapján elvégezte.

Debrecen, 2022.09.22.

DEBRECENI EGYETEM Egyetemi és Nemzeti Könyvtár

Bibliography

[1] ImageNet. URL http://www.image-net.org

[2] Abásolo, D., Hornero, R., Espino, P., Poza, J., Sánchez, C.I., de la Rosa, R.: Analysis of regularity in the EEG background activity of Alzheimer's disease patients with Approximate Entropy. Clinical Neurophysiology **116**(8), 1826–1834 (2005)

[3] Andrews, J.T.A., Tanay, T., Morton, E.J., Griffin, L.D.: Transfer representation-learning for anomaly detection. ICML (2016)

[4] Antal, B., Hajdu, A.: An ensemble-based system for microaneurysm detection and diabetic retinopathy grading. IEEE Trans. on Biomed. Eng. **59**(6), 1720–1726 (2012). DOI 10.1109/TBME.2012.2193126

[5] Antal, B., Hajdu, A.: An ensemble-based system for automatic screening of diabetic retinopathy. Knowledge-Based Systems **60**, 20–27 (2014). DOI https://doi.org/10.1016/j.knosys.2013.12.023

[6] Baldoni, V., Berline, N., Vergne, M.: Summing a polynomial function over integral points of a polygon, User's guide. HAL **2009**(0) (2009). URL http://dml.mathdoc.fr/item/hal-00383196

[7] Barvinok, A.I.: A polynomial time algorithm for counting integral points in polyhedra when the dimension is fixed. 34th Annual Sym-

posium of Foundations of Computer Science, IEEE pp. 566–572 (1993)

[8] Bernardin, L., Chin, P., DeMarco, P., Geddes, K.O., Harea, D.E.G., Heal, K.M., Labahn, G., May, J.P., Monagan, J.M.M.B., Ohashi, D., Vorkoetter, S.M.: Maple Programming Guide. Maplesoft, a division of Waterloo Maple Inc. (1996-2021)

[9] Bhimala, K.R., Patra, G.K., Mopuri, R., Mutheneni, S.R.: Prediction of COVID-19 cases using the weather integrated deep learning approach for India. Transboundary and Emerging Diseases **69**(3), 1349–1363 (2022). DOI https://doi.org/10.1111/tbed.14102

[10] Bucilu, C., Caruana, R., Niculescu-Mizil, A.: Model compression. In: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '06, p. 535–541. Association for Computing Machinery, New York, NY, USA (2006). DOI 10.1145/1150402.1150464

[11] Cavalcanti, G.D., Oliveira, L.S., Moura, T.J., Carvalho, G.V.: Combining diversity measures for ensemble pruning. Patt. Rec. Lett. **74**, 38–45 (2016). DOI https://doi.org/10.1016/j.patrec.2016.01.029

[12] Chalapathy, R., Menon, A., Chawla, S.: Anomaly detection using one-class neural networks. ArXiv **abs/1802.06360** (2018)

[13] Cho, S.B., Kim, J.H.: Combining multiple neural networks by fuzzy integral for robust classification. IEEE Transactions on Systems, Man, and Cybernetics **25**(2), 380–384 (1995). DOI 10.1109/21.364825

[14] Cohen, H.: A course in computational number theory, third corrected printing edn. Springer (1996)

[15] Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation **6**(2), 182–197 (2002)

[16] Dheeru, D., Karra Taniskidou, E.: UCI machine learning repository. University of California, Irvine, School of Information and Computer Sciences (2017). URL http://archive.ics.uci.edu/ml

[17] Du, K., Swamy, M.: Search and Optimization by Metaheuristics: Techniques and Algorithms Inspired by Nature. Springer International Publishing (2016)

[18] Freund, Y., Schapire, R.E.: Large margin classification using the perceptron algorithm. Machine Learning **37**, 277–296 (1999)

[19] Friedman, N., Geiger, D., Goldszmidt, M.: Bayesian network classifiers. Machine Learning **29**(2), 131–163 (1997)

[20] Gadkari, D.: Image quality analisys using GLCM. Electronic Theses and Dissertations (187), University of Central Florida, Orlando. (2004). URL https://stars.library.ucf.edu/etd/187

[21] Ghany, K.K.A., Zawbaa, H.M., Sabri, H.M.: COVID-19 prediction using LSTM algorithm: GCC case study. Informatics in Medicine Unlocked **23**, 100566 (2021). DOI https://doi.org/10.1016/j.imu.2021.100566

[22] Goldberg, D.: Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley Publishing (1989)

[23] Gonzalez, M.C., Hidalgo, C.A., Barabási, A.L.: Understanding individual human mobility patterns. Nature **453**(7196), 779–782 (2008)

[24] Gonzalez, R.C., Woods, R.E.: Digital image processing, pp. 90–177. Prentice Hall, Upper Saddle River, N.J. (2008)

[25] Hajdu, A., Hajdu, L., Jónás, A., Kovács, L., Tomán, H.: Generalizing the majority voting scheme to spatially constrained voting. IEEE Transactions on Image Processing **22**(11), 4182–4194 (2013). DOI 10.1109/TIP.2013.2271116

[26] Hajdu, A., Hajdu, L., Kovács, L., Tomán, H.: Diversity measures for majority voting in the spatial domain. In: J.S. Pan, M.M. Polycarpou, M. Woźniak, A.C.P.L.F. de Carvalho, H. Quintián, E. Corchado (eds.) Hybrid Artificial Intelligent Systems, pp. 314–323. Springer Berlin Heidelberg, Berlin, Heidelberg (2013)

[27] Hajdu, A., Hajdu, L., Tijdeman, R.: Finding well approximating lattices for a finite set of points. MATH COMPUT **2017**, 1–24 (2017)

[28] Hajdu, A., Harangi, B., Besenczi, R., Lázár, I., Emri, G., Hajdu, L., Tijdeman, R.: Measuring regularity of network patterns by grid approximations using the LLL algorithm. In: 23rd International Conference on Pattern Recognition (ICPR 2016), pp. 1525–1530. Cancun, Mexico (2016)

[29] Hajdu, A., Terdik, G., Tiba, A., Toman, H.: A stochastic approach to handle resource constraints as knapsack problems in ensemble pruning. Machine Learning **111**, 1551–1595 (2022). DOI 10.1007/s10994-021-06109-0

[30] Hajdu, A., Tomán, H., Kovács, L., Hajdu, L.: Composing ensembles by a stochastic approach under execution time constraint. In: 2016 23rd International Conference on Pattern Recognition (ICPR), pp. 222–227 (2016). DOI 10.1109/ICPR.2016.7899637

[31] Hajdu, L., Harangi, B., Tiba, A., Hajdu, A.: Detecting periodicity in digital images by the LLL algorithm. In: I. Faragó, F. Izsák, P.L.

Simon (eds.) Progress in Industrial Mathematics at ECMI 2018, pp. 613–619. Springer International Publishing, Cham (2019)

[32] Harangi, B.: Skin lesion classification with ensembles of deep convolutional neural networks. Journal of Biomedical Informatics **86**, 25–32 (2018)

[33] Harangi, B., Baran, A., Hajdu, A.: Classification of skin lesions using an ensemble of deep neural networks. In: 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBC 2018, Honolulu, HI, USA, July 18-21, 2018, pp. 2575–2578 (2018)

[34] He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778. IEEE (2016). DOI https://doi.org/10.1109/CVPR.2016.90

[35] Hernández-Lobato, D., Martinez-Munoz, G., Suarez, A.: Statistical instance-based pruning in ensembles of independent classifiers. IEEE Transactions on Pattern Analysis and Machine Intelligence **31**(2), 364–369 (2009)

[36] Hethcote, H.: Modeling heterogeneous mixing in infectious disease dynamics, p. 215–238. Publications of the Newton Institute. Cambridge University Press (1996). DOI 10.1017/CBO9780511662935.030

[37] Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network (2015). URL http://arxiv.org/abs/1503.02531

[38] Ho, T.K., Hull, J.J., Srihari, S.N.: Decision combination in multiple classifier systems. IEEE Transactions on Pattern Analysis and Machine Intelligence **16**(1), 66–75 (1994). DOI 10.1109/34.273716

[39] Hochreiter, S., Schmidhuber, J.: Long Short-Term Memory. Neural Computation **9**(8), 1735–1780 (1997). DOI 10.1162/neco.1997.9.8.1735

[40] Huang, Y.S., Suen, C.Y.: A method of combining multiple experts for the recognition of unconstrained handwritten numerals. IEEE Transactions on Pattern Analysis and Machine Intelligence **17**(1), 90–94 (1995). DOI 10.1109/34.368145

[41] John, G.H., Langley, P.: Estimating continuous distributions in bayesian classifiers. In: Eleventh Conference on Uncertainty in Artificial Intelligence, pp. 338–345. San Mateo (1995)

[42] Johns Hopkins University: New cases of COVID-19 in world countries. URL https://coronavirus.jhu.edu/data/new-cases. Accessed 12th Apr 2020

[43] Kim, C.E.: On the cellular convexity of complexes. IEEE Transactions on Pattern Analysis and Machine Intelligence **PAMI-3(6)**, 617–625 (1981)

[44] Kim, C.E., Rosenfeld, A.: Digital straight lines and convexity of digital regions. IEEE Transactions on Pattern Analysis and Machine Intelligence **PAMI-4(2)**, 149–153 (1982)

[45] Klastorin, T.: On a discrete nonlinear and nonseparable knapsack problem. Operations Research Letters **9**(4), 233 – 237 (1990). DOI https://doi.org/10.1016/0167-6377(90)90067-F

[46] Kolozsvári, L.R., Bérczes, T., Hajdu, A., Gesztelyi, R., Tiba, A., Varga, I., Al-Tammemi, A.B., Szőllősi, G.J., Harsányi, S., Garbóczy, S., Zsuga, J.: Predicting the epidemic curve of the coronavirus (SARS-CoV-2) disease (COVID-19) using artificial in-

telligence: An application on the first and second waves. Informatics in Medicine Unlocked **25**, 100691 (2021). DOI https://doi.org/10.1016/j.imu.2021.100691

[47] Kong, E.B., Dietterich, T.G.: Error-correcting output coding corrects bias and variance. In: A. Prieditis, S. Russell (eds.) Machine Learning Proceedings 1995, pp. 313–321. Morgan Kaufmann, San Francisco (CA) (1995). DOI https://doi.org/10.1016/B978-1-55860-377-6.50046-3

[48] Krizhevsky, A., Sutskever, I., Hinton, G.: Imagenet classification with deep convolutional neural networks. Communications of the ACM **60**(6), 1079–1105 (2017). DOI https://doi.org/10.1145/3065386

[49] Krupic, J., Burgess, N., OḰeefe, J.: Neural representations of location composed of spatially periodic bands. Science **337(6096)**, 853–857 (2012)

[50] Kuncheva, L.I.: Combining Pattern Classifiers: Methods and Algorithms. Wiley-Interscience (2004)

[51] Kurz, M., Hölzl, G., Ferscha, A.: Enabling dynamic sensor configuration and cooperation in opportunistic activity recognition systems. International Journal of Distributed Sensor Networks **9**(6), 652385 (2013). DOI 10.1155/2013/652385

[52] İsmail Kırbaş, Sözen, A., Tuncer, A.D., Şinasi Kazancıoğlu, F.: Comparative analysis and forecasting of COVID-19 cases in various european countries with ARIMA, NARNN and LSTM approaches. Chaos, Solitons and Fractals **138**, 110015 (2020). DOI https://doi.org/10.1016/j.chaos.2020.110015

[53] Lam, L., Suen, S.Y.: Application of majority voting to pattern recognition: An analysis of its behavior and performance. Trans. Sys. Man Cyber. Part A **27**(5), 553–568 (1997). DOI 10.1109/3468.618255

[54] Larochelle, H., Bengio, Y.: Classification using discriminative restricted Boltzmann machines. Proceedings of the 25th International Conference on Machine Learning (ICML) pp. 536–543 (2008)

[55] LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE **86**(11), 2278–2324 (1998)

[56] Lenstra, A.K., Lenstra, W., Lovász, L.: Factoring polynomials with rational coefficients. Math. Ann **261**, 515–534 (1982)

[57] Luo, J., Zhang, Z., Fu, Y., Rao, F.: Time series prediction of COVID-19 transmission in america using LSTM and XGBoost algorithms. Results in Physics **27**, 104462 (2021). DOI https://doi.org/10.1016/j.rinp.2021.104462

[58] Martello, S., Toth, P.: Knapsack Problems: Algorithms and Computer Implementations. John Wiley & Sons, Inc., New York, NY, USA (1990)

[59] Martinez-Munoz, G., Suarez, A.: Using boosting to prune bagging ensembles. Pattern Recognition Letters **28**, 156–165 (2007). DOI https://doi.org/10.1016/j.patrec.2006.06.018

[60] Mousavi, R., Eftekhari, M.: A new ensemble learning methodology based on hybridization of classifier ensemble selection approaches. Applied Soft Computing **37**, 652 – 666 (2015). DOI https://doi.org/10.1016/j.asoc.2015.09.009

[61] Neumann, J., Schnorr, C., Steidl, G.: Combined SVM-based feature selection and classification. Machine learning **61**(1-3), 129–150 (2005)

[62] Nishani, E., Cico, B.: Computer vision approaches based on deep learning and neural networks: Deep neural networks for video analysis of human pose estimation. In: 6th Mediterranean Conference on Embedded Computing (MECO), pp. 1–4 (2017)

[63] Nixon, M., Aguado, A.: Feature Extraction and Image Processing, 2nd edn. Academic Press is an imprint of Elsevier, Oxford, UK (2008)

[64] Palaz, D., Magimai-Doss, M., Collobert, R.: Analysis of CNN-based speech recognition system using raw speech as input. In: Proc. Interspeech 2015, pp. 11–15. InterSpeech (2015). DOI 10.21437/Interspeech.2015-3

[65] Patsadu, O., Nukoolkit, C., Watanapa, B.: Human gesture recognition using kinect camera. In: Ninth International Conference on Computer Science and Software Engineering, pp. 28 – 31 (2012)

[66] Quinlan, J.R.: Induction of decision trees. Machine Learning **1**, 81–106 (1986)

[67] Rahele, K., Roya, A., Narges, S., Zahra, A., Nasim Dadashi, S., Shervin, M., Sunil Kumar, Y., Atefeh, V., Nima, R., Shaghayegh Haghjooy, J.: COVID-19 in Iran: Forecasting pandemic using deep learning. Computational and Mathematical Methods in Medicine p. 16 (2021). DOI https://doi.org/10.1155/2021/6927985

[68] Scholkopf, B., Smola, A.J.: Support vector machines, regularization, optimization, and beyond. MIT Press (2002)

[69] Sharkey, T.C., Romeijn, H.E., Geunes, J.: A class of nonlinear non-separable continuous knapsack and multiple-choice knapsack problems. Mathematical Programming **126**(1), 69–96 (2011). DOI 10.1007/s10107-009-0274-9

[70] Shiraishi, Y., Fukumizu, K.: Statistical approaches to combining binary classifiers for multi-class classification. Neurocomput. **74**(5), 680–688 (2011). DOI 10.1016/j.neucom.2010.09.004

[71] Siettos, C.I., Russo, L.: Mathematical modeling of infectious disease dynamics. Virulence **4**(4), 295–306 (2013). DOI 10.4161/viru.24041. PMID: 23552814

[72] Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: International Conference on Learning Representations, p. 1 (2015)

[73] Suratgar, A.A., Tavakoli, M.B., Hoseinabadi, A.: Modified Levenberg-Marquardt method for neural networks training. World Academy of Science, Engineering and Technology p. 24–48 (2005)

[74] Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. CoRR **abs/1409.3215** (2014). URL http://arxiv.org/abs/1409.3215

[75] Suykens, J.A.K., Vandewalle, J.: Training multilayer perceptron classifiers based on a modified support vector method. IEEE Transaction on Neural Networks **10**(4) (1999)

[76] Szegedy, C., Liu, W., Jia, Y., Jia, Y., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–9. IEEE (2015). DOI https://doi.org/10.1109/cvpr.2015.7298594

[77] Tang J.and Gupta, A.K.: On the distribution of the product of independent beta random variables. Statistics & Probability Letters **2**(3), 165–168 (1984)

[78] Tempo, R., Ishii, H.: Monte Carlo and Las Vegas randomized algorithms for systems and control*: An introduction. European Journal of Control **13**(2), 189–203 (2007). DOI https://doi.org/10.3166/ejc.13.189-203

[79] Tiba, A., Bartik, Z., Toman, H., Hajdu, A.: Detecting outlier and poor quality medical images with an ensemble-based deep learning system. In: 2019 11th International Symposium on Image and Signal Processing and Analysis (ISPA), pp. 99–104 (2019). DOI 10.1109/ISPA.2019.8868911

[80] Tiba, A., Hajdu, A., Terdik, G., Tomán, H.: Optimizing majority voting based systems under a resource constraint for multiclass problems. In: I. Faragó, F. Izsák, P.L. Simon (eds.) Progress in Industrial Mathematics at ECMI 2018, pp. 529–534. Springer International Publishing, Cham (2019)

[81] Tiba, A., Harangi, B., Hajdu, A.: Efficient texture regularity estimation for second order statistical descriptors. In: Proceedings of the 10th International Symposium on Image and Signal Processing and Analysis, pp. 90–94 (2017). DOI 10.1109/ISPA.2017.8073575

[82] Timotheou, S.: The random neural network: A survey. Comput. J. **53**, 251–267 (2010)

[83] Tomar, A., Gupta, N.: Prediction for the spread of COVID-19 in India and effectiveness of preventive measures. Science of The Total Environment **728**, 138762 (2020). DOI https://doi.org/10.1016/j.scitotenv.2020.138762

[84] Utsumi, T., Iwatate, M., Sano, W., Sunakawa, H., Hattori, S., Hasuike, N., Sano, Y.: Polyp Detection, Characterization, and Management Using Narrow Band Imaging with/without Magnification. Gastrointestinal Center and Institution of Minimally Invasive Endoscopic Care (iMEC), Sano Hospital, Kobe, Japan (2015)

[85] Williamson, D.P., Shmoys, D.B.: The Design of Approximation Algorithms, 1st edn. Cambridge University Press, USA (2011)

[86] Wu, Z., McGoogan, J.M.: Characteristics of and Important Lessons From the Coronavirus Disease 2019 (COVID-19) Outbreak in China: Summary of a Report of 72314 Cases From the Chinese Center for Disease Control and Prevention. JAMA **323**(13), 1239–1242 (2020). DOI 10.1001/jama.2020.2648

[87] Xianglei, Z., Fu, B., Yang, Y., Ma, Y., Hao, J., Chen, S., Liu, S., Li, T., Liu, S., Guo, W., Liao, Z.: Attention-based recurrent neural network for influenza epidemic prediction. BMC Bioinformatics **20**, 575 (2019). DOI 10.1186/s12859-019-3131-8

[88] Xu, L., Krzyzak, A., Suen, C.Y.: Methods of combining multiple classifiers and their applications to handwriting recognition. IEEE Transactions on Systems, Man, and Cybernetics **22**(3), 418–435 (1992). DOI 10.1109/21.155943

[89] Zhou, Z.H.: Ensemble Methods: Foundations and Algorithms, 1st edn. Chapman & Hall/CRC (2012)

[90] Zhu, R., Zhang, R., Xue, D.: Lesion detection of endoscopy images based on convolutional neural network features. International Congress on Image and Signal Processing pp. 372–376 (2015)

Appendices

4.3 Proof of Lemma 1.

Proof. Consider a subset $\mathcal{K}$ when $\mathcal{K} = \{1, 2, \ldots, 2\ell\}$ (otherwise we can renumerate p_i). We have

$$q_{2\ell}(\mathcal{K}) = \sum_{k=\ell+1}^{2\ell} \sum_{\substack{I \subseteq \mathcal{K} \\ |I|=k}} \prod_{i \in I} p_i \prod_{j \in \mathcal{K} \setminus I} (1 - p_j) = \sum_{k=\ell+1}^{2\ell} \sum_{\substack{I \subseteq \mathcal{K} \\ |I|=k}} Q_{2\ell,k}(\mathcal{K}, I), \tag{4.1}$$

where

$$Q_{2\ell,k}(\mathcal{K}, I) = \prod_{i \in I} p_i \prod_{j \in \mathcal{K} \setminus I} (1 - p_j), \tag{4.2}$$

that is, we consider a subset $\mathcal{K} \subseteq \mathcal{N}$ with $|\mathcal{K}| = 2\ell$, and $Q_{2\ell,k}(\mathcal{K}, I)$ is calculated for an index set $I \subseteq \mathcal{K}$ with $|I| = k$. Now, choose an index a from the set $\mathcal{N} \setminus \mathcal{K}$, i.e., $a > 2\ell$, and obtain

$$Q_{2\ell,k}(\mathcal{K}, I) = Q_{2\ell,k}(\mathcal{K}, I)\, p_a + Q_{2\ell,k}(\mathcal{K}, I)\,(1 - p_a). \tag{4.3}$$

The term $Q_{2\ell,k}\left(\mathcal{K},I\right)p_a = Q_{2\ell+1,k+1}\left(\{\mathcal{K},a\},\{I,a\}\right)$
and $Q_{2\ell,k}\left(\mathcal{K},I\right)\left(1-p_a\right) = Q_{2\ell+1,k}\left(\{\mathcal{K},a\},I\right)$; therefore,

$$q_{2\ell}\left(\mathcal{K}\right) = \sum_{k=\ell+1}^{2\ell}\sum_{\substack{I\subseteq\mathcal{K}\\|I|=k}} Q_{2\ell,k}\left(\mathcal{K},I\right) =$$

$$= \sum_{k=\ell+1}^{2\ell}\left(\sum_{\substack{I\subseteq\mathcal{K}\\|I|=k}} Q_{2\ell+1,k+1}\left(\{\mathcal{K},a\},\{I,a\}\right) + \sum_{\substack{I\subseteq\mathcal{K}\\|I|=k}} Q_{2\ell+1,k}\left(\{\mathcal{K},a\},I\right)\right) <$$

$$< \sum_{k=\ell+1}^{2\ell}\sum_{\substack{I\subseteq\mathcal{K}\\|I|=k}} Q_{2\ell+1,k}\left(\mathcal{K},I\right) = q_{2\ell+1}\left(\mathcal{K},a\right), \tag{4.4}$$

since $q_{2\ell+1}\left(\mathcal{K}\right)$ includes some extra additional terms, say $Q_{2\ell+1,k}\left(\{\mathcal{K},a\},I\right)$, where I contains a. Regarding that n is odd in the series of $q_{\ell}\left(\mathcal{K}\right)$, there will be an element with odd ℓ following an element of even ℓ and the lemma is proved for odd n. For the case when n is even, we consider the $q_n\left(\mathcal{N}\right)$ and $q_{n-1}\left(\mathcal{L}\right)$, where $\mathcal{L} = \{1,2,\ldots,2\ell-1\}$. Set $n = 2\ell$; then,

$$q_{2\ell}\left(\mathcal{N}\right) = \sum_{k=\ell+1}^{2\ell}\sum_{\substack{I\subseteq\mathcal{N}\\|I|=k}} Q_{2\ell,k}\left(\mathcal{N},I\right) \tag{4.5}$$

with $\mathcal{N} = \{1,2,\ldots,2\ell\}$ and put

$$q_{2\ell-1}\left(\mathcal{L}\right) = \sum_{k=\ell}^{2\ell-1}\sum_{\substack{I\subseteq\mathcal{L}\\|I|=k}} Q_{2\ell-1,k}\left(\mathcal{L},I\right), \tag{4.6}$$

notice that the number of terms is equal in both sums. If $k = 2\ell$, then

$$Q_{2\ell,2\ell}\left(\mathcal{N},\mathcal{N}\right) = p_{2\ell}Q_{2\ell-1,2\ell-1}\left(\mathcal{L},\mathcal{L}\right), \tag{4.7}$$

otherwise,

$$Q_{2\ell,k}\left(\mathcal{N},I\right) = \begin{cases} p_{2\ell}Q_{2\ell-1,k-1}\left(\mathcal{L},I\backslash 2\ell\right) & \text{if} \quad 2\ell\in I \\ \left(1-p_{2\ell}\right)Q_{2\ell-1,k}\left(\mathcal{L},I\right) & \text{if} \quad 2\ell\notin I \end{cases}, \tag{4.8}$$

hence for $k < 2\ell$,

$$\sum_{\substack{I\subseteq\mathcal{N}\\|I|=k}} Q_{2\ell,k}\left(\mathcal{N},I\right) = p_{2\ell} \sum_{\substack{I\subseteq\mathcal{L}\\|I|=k-1}} Q_{2\ell-1,k-1}\left(\mathcal{L},I\right) + \left(1-p_{2\ell}\right) \sum_{\substack{I\subseteq\mathcal{L}\\|I|=k}} Q_{2\ell-1,k}\left(\mathcal{L},I\right). \tag{4.9}$$

We start summing up $q\left(\mathcal{N},2\ell\right)$ from 2ℓ; then, using (4.7) and (4.9), we obtain for the first two terms

$$\begin{aligned} &\sum_{k=2\ell-1}^{2\ell} \sum_{\substack{I\subseteq\mathcal{N}\\|I|=2\ell-1}} Q_{2\ell,k}\left(\mathcal{N},I\right) = p_{2\ell} Q_{2\ell-1,2\ell-1}\left(\mathcal{L},\mathcal{L}\right) \\ &+p_{2\ell} \sum_{\substack{I\subseteq\mathcal{L}\\|I|=2\ell-2}} Q_{2\ell-1,2\ell-2}\left(\mathcal{L},I\right) + \left(1-p_{2\ell}\right) Q_{2\ell-1,2\ell-1}\left(\mathcal{L},\mathcal{L}\right) \\ &= Q_{2\ell-1,2\ell-1}\left(\mathcal{L},\mathcal{L}\right) + p_{2\ell} \sum_{\substack{I\subseteq\mathcal{L}\\|I|=2\ell-2}} Q_{2\ell-1,2\ell-2}\left(\mathcal{L},I\right). \end{aligned} \tag{4.10}$$

If we continue summing up one by one, then induction leads to

$$q_{2\ell}\left(\mathcal{N}\right) = \sum_{k=\ell+1}^{2\ell-1} \sum_{\substack{I\subseteq\mathcal{L}\\|I|=k}} Q_{2\ell-1,k-1}\left(\mathcal{L},I\right) + p_{2\ell} \sum_{\substack{I\subseteq\mathcal{L}\\|I|=\ell}} Q_{2\ell-1,\ell}\left(\mathcal{L},I\right) < q_{2\ell-1}\left(\mathcal{L}\right), \tag{4.11}$$

if $p_{2\ell} < 1$. If $p_{2\ell} = 1$ then $q_{2\ell} = q_{2\ell-1}$.

□

4.4 Proof of Proposition 2.3.1.

Proof. We prove the statement with an example describing the worst-case scenario for the greedy selection strategy. Let $\mathcal{D} = \{D_1 = (p_1, t_1), D_2 = (p_2, t_2), \ldots, D_n = (p_n, t_n)\}$ be the pool, where the index set is denoted by $\mathcal{I}_n = \{1, 2, \ldots, n\}$. Let us suppose that $\sum_{i=1}^{n} t_i \leq T$, that is, the time constraint should not be of concern. Let $p_1 = 1/2+\varepsilon$, where $0 < \varepsilon < 1/2$,

and $p_2 = p_3 = \cdots = p_n = 1/2 + \alpha$ with $0 < \alpha < \varepsilon$, where the proper selection of α will be given below.

The greedy strategy will move D_1 to S as the most accurate item in its first step. Next, we try to extend S by adding more members. Since we require odd members, we try to add 2 items in every selection step. Since all the remaining $n - 1$ features have the same behavior, we can check whether S should be extended via comparing the performance of $S_1 = \{D_1\}$ and $S_3 = \{D_1, D_2, D_3\}$. For the performance of the ensemble S_1, we trivially have $q_1(\mathcal{I}_1) = p_1 = 1/2 + \varepsilon$, where $\mathcal{I}_1 = \{1\}$, while for S_3 we can apply (2.1) for the 3-member ensemble, with $\mathcal{I}_3 = \{1, 2, 3\}$ to calculate $q_3(\mathcal{I}_3)$:

$$\begin{aligned} q_3(\mathcal{I}_3) \quad &= p_1p_2(1-p_3) + p_2p_3(1-p_1) + p_1p_3(1-p_2) + \\ &\quad + p_1p_2p_3 = \tfrac{1}{2} + \tfrac{\varepsilon}{2} + \alpha - 2\alpha^2\varepsilon \end{aligned} \tag{4.12}$$

after the appropriate substitutions and simplifications. Now, if we adjust α to have $q_1(\mathcal{I}_1) = q_3(\mathcal{I}_3)$, then via solving the equation

$$\frac{1}{2} + \varepsilon = \frac{1}{2} + \frac{\varepsilon}{2} + \alpha - 2\alpha^2\varepsilon \tag{4.13}$$

we obtain

$$\alpha = \frac{1 - \sqrt{1 - 4\varepsilon^2}}{4\varepsilon}. \tag{4.14}$$

That is, with a selection of α given in (4.14), the ensemble $S_1 = \{D_1\}$ is not going to be extended since it does not lead to improvement. Thus, the strategy stops after the first step with an ensemble accuracy $1/2 + \varepsilon$.

On the other hand, with a sufficiently large n, a very accurate ensemble could be achieved. More precisely, it can be easily seen that $q_n(\mathcal{I}_n)$ is strictly monotonically increasing with

$$\lim_{n\to\infty} q_n(\mathcal{I}_n) = 1. \tag{4.15}$$

Now, by letting $\varepsilon \to 0$, we can see that for the ensemble accuracy found with this strategy

$$\lim_{\varepsilon \to 0} q_1(S_1) = 1/2, \tag{4.16}$$

while an ensemble of $\lim_{n \to \infty} q_n(\mathcal{I}_n) = 1$ could also be found. Hence, the proposition follows. □

4.5 Proof of Proposition 2.3.2.

Proof. We prove the statement with a similar example to that given in the proof of Proposition 2.3.1 in Appendix 4.4 to describe the worst-case scenario. Let $\mathcal{D} = \{D_1 = (p_1, t_1), D_2 = (p_2, t_2), \ldots, D_n = (p_n, t_n)\}$ be the pool and T be the time constraint. Put $p_1 = 1/2 + \varepsilon$, where $0 < \varepsilon \leq 1/2$, $t_1 = T$, and $p_2 = p_3 = \cdots = p_m = 1/2 + \alpha$, $t_2 = t_3 = \cdots = t_n = T/(n-1)$ with $0 < \alpha < \varepsilon$. If α is properly selected, then $q_1(\mathcal{I}_1) = p_1 < q_{n-1}(\mathcal{I}_n \setminus \mathcal{I}_1)$. However, because of the time constraint, we must remove elements during the selection procedure, since initially $\sum_{i=1}^{n} t_i = 2T > T$. For this requirement, the greedy approach in the first step will remove any two elements from $D_2, \ldots, D_n$ by decreasing the time with $2T/(n-1)$. This selection will go on until only D_1 remains in the ensemble. With a proper selection of α, we have $\lim_{n \to \infty} q_{n-1}(\mathcal{I}_n \setminus \mathcal{I}_1) = 1$ and by letting $\varepsilon \to 0$, the proposition follows. □

4.6 Proof of Proposition 2.3.3.

Proof. Similar to the proof of Proposition 2.3.2, we provide an example for the worst-case scenario. Let $D_1 = (1, T)$, and $D_2 = D_3 = D_4 = (1/2 + \varepsilon, T/3)$ with $0 < \varepsilon < 1/2$. Now, since

$$u_1 = \frac{1}{T} < \frac{3/2 + 3\varepsilon}{T} = u_2 = u_3 = u_4, \tag{4.17}$$

the backward strategy will remove the less useful component D_1 first to maintain the time constraint and will keep the remaining ensemble $\{D_2, D_3, D_4\}$ as the most accurate one, which also fits the time constraint with $\sum_{i=2}^{4} t_i = T$. By letting $\varepsilon \to 0$, we have $\lim_{\varepsilon \to 0} q_3(\mathcal{I}_4 \setminus \mathcal{I}_1) = 1/2$. Moreover, notice that the most accurate ensemble would have been $\{D_1\}$ with $q_1(\mathcal{I}_1) = 1$ by meeting the time constraint as well. Thus, the statement follows. □

4.7 Lemma 2.

Lemma 2. *Let $p \in [0,1]$ be a random variable with mean μ_p and variance σ_p^2. Consider the accuracy (2.1), where p_i, $i = 1, 2, \ldots, n$ are i.i.d. random variables distributed as p. Then,*

1.

$$\lim_{\ell \to \infty} \mu_{q_\ell} = \begin{cases} 0, & \text{if} \quad \mu_p \notin [1/2, 1), \\ 1/2, & \text{if} \quad \mu_p = 1/2, \\ 1, & \text{if} \quad \mu_p \in (1/2, 1). \end{cases} \tag{4.18}$$

Moreover, for odd ℓ: if $\mu_p \in (1/2, 1)$, then μ_{q_ℓ} is increasing, and if $\mu_p \in (0, 1/2)$, then μ_{q_ℓ} is decreasing.

2. The variance of q_ℓ is expressed by

$$\begin{aligned} \sigma_{q_\ell}^2 = \sum_{k=k_\ell}^{\ell} \sum_{m=1}^{k} \sum_{h=k_\ell - m}^{\ell - k} \delta(\ell, m, k) \binom{\ell}{k} \\ \times \binom{k}{m} \binom{\ell - k}{h} s_T^m s_{TF}^{k-m+h} s_F^{\ell-k-h} - (\mu_{q_\ell})^2, \end{aligned} \tag{4.19}$$

where $\delta(\ell, m, k) = \delta_{k_\ell - m \le \ell - k}$, $s_T = \sigma_p^2 + \mu_p^2$, $s_F = \sigma_p^2 + (1 - \mu_p)^2$, $s_{TF} = \mu_p(1 - \mu_p) - \sigma_p^2$, and $k_\ell = \left\lfloor \frac{\ell}{2} \right\rfloor + 1$.

3. If $\mu_p \neq 1/2$, $\mu_p(1-\mu_p) - \sigma_p^2 > 0$, *and* $s_T \neq 1/2$, *then*

$$\lim_{\ell \to \infty} \sigma_{q_\ell}^2 = 0. \tag{4.20}$$

If $s_T = 1/2$, *then the limit (4.20) is 1.*

Proof. The first part of the lemma corresponds to Theorem 1 in [53]. For the rest, let us denote the product of probabilities by

$$\Pi(I) = \prod_{i \in I} p_i \prod_{j \in \mathcal{N} \setminus I} (1 - p_j), \tag{4.21}$$

for simplifying the treatment below. The formula (4.19) follows from expressing the variance in terms of covariance

$$\mathrm{Var}(q_\ell) = \sum_{k,j=k_\ell}^{\ell} \sum_{\substack{I,J \subseteq \mathcal{N} \\ |I|=k, |J|=j}} \mathrm{Cov}(\Pi(I), \Pi(J)). \tag{4.22}$$

Now, we rewrite this expression into a more appropriate form. First, the notation is introduced, where I_T^k and I_F^k for a partition of indices $\mathcal{N} = \{1, \ldots, \ell\}$, such that $\mathcal{N} = I_T^k \cup I_F^k$ where I_T^k denotes indices of those members voting true with accuracy p. Similarly, I_F^k contains indices of false votes. Observe $I_F^k = \mathcal{N} \setminus I_T^k$. We have $\left|I_T^k\right| = k$ and $\left|I_F^k\right| = \ell - k$. In the case of two partitions $I_T^k \cup I_F^k$ and $J_T^j \cup J_F^j$, let the number of the common elements of I_T^k and J_T^j be $\left|I_T^k \cap J_T^j\right| = n_{k.j}$; similarly, $\left|I_F^k \cap J_F^j\right| = m_{k.j}$. According to this setup

$$\mathrm{Var}(q_\ell) = \sum_{k,j=k_\ell}^{\ell} \sum_{I_T^k, J_T^j} \mathrm{Cov}\left(\Pi\left(I_T^k\right), \Pi\left(J_T^k\right)\right). \tag{4.23}$$

Observe $I_F^k = \mathcal{N} \setminus I_T^k$ when we apply the notation for the product. Now, we consider the covariance

$$\begin{aligned} &\mathrm{Cov}\left(\Pi\left(I_T^k\right), \Pi\left(J_T^k\right)\right) \\ &\quad = \mathrm{E}\,\Pi\left(I_T^k\right)\Pi\left(J_T^k\right) - \mathrm{E}\,\Pi\left(I_T^k\right)\mathrm{E}\,\Pi\left(J_T^k\right) \\ &\quad = \mathrm{E}\,\Pi\left(I_T^k\right)\Pi\left(J_T^k\right) - \mu^{k+j}(1-\mu)^{2\ell-k-j}. \end{aligned} \tag{4.24}$$

The first term contains three types of products:

$$\mathrm{E}\, p^2 = s_T = \sigma_p^2 + \mu_p^2, \tag{4.25}$$

$$\mathrm{E}\,(1-p)^2 = s_F = \sigma_p^2 + (1-\mu_p)^2\,, \tag{4.26}$$

$$\mathrm{E}\, p\,(1-p) = s_{TF} = \mu_p\,(1-\mu_p) - \sigma_p^2. \tag{4.27}$$

The pool constitutes independent variables; therefore,

$$\begin{aligned}\mathrm{Var}\,(q_\ell) = \sum_{k,j=k_\ell}^{\ell} \sum_{I^k,J^j} \left(\sigma_p^2 + \mu_p^2\right)^{n_{k.j}} \left(\sigma_p^2 + (1-\mu_p)^2\right)^{m_{k.j}} \\ \times \left(\mu_p\,(1-\mu_p) - \sigma_p^2\right)^{\ell - n_{k.j} - m_{k.j}} - (Eq_\ell)^2\,. \end{aligned} \tag{4.28}$$

since the sum of the second term gives the $(Eq_\ell)^2$, indeed

$$\begin{aligned}\sum_{k,j=k_\ell}^{\ell} \binom{\ell}{k}\binom{\ell}{j} \mu_p^{k+j}\,(1-\mu_p)^{2\ell-k-j} \\ = \left(\sum_{k=k_\ell}^{\ell} \binom{\ell}{k} \mu_p^k\,(1-\mu_p)^{\ell-k}\right)^2 = (Eq_\ell)^2\,. \end{aligned} \tag{4.29}$$

We simplify (4.28), collecting similar terms and obtain (4.19). Before we prove the limit (4.20), let us observe

$$s_T + s_{TF} = \mu_p, \tag{4.30}$$

$$s_F + s_{TF} = 1 - \mu_p, \tag{4.31}$$

$$s_T + s_F + 2s_{TF} = 1. \tag{4.32}$$

i.e., the set $\{s_T, s_{TF}, s_F, s_{TF}\}$ constitutes a probability distribution for $s_{TF} > 0$; in other words, $\mu_p^2 + \sigma_p^2 < \mu_p$. If it is so, we rewrite (4.19) in the form of a multinomial distribution. The coefficients in (4.19) are actually multinomial coefficients. The rest of the proof is based on the approximation of the binomial distribution by the normal distribution. It is not complicated but slightly lengthy; we make it available to the interested readers on request. □

4.8 Lemma 3.

Lemma 3. *Let τ be an exponential distribution with density $\lambda \exp(-\lambda t)$ under the condition that the parameter λ is distributed as Beta(β_p, α_p), where $2 < \beta_p < \alpha_p$.*

1. *Then, the expected time for the sum of n variables is*

$$\sum_{k=0}^{n} E\tau_k = n\left(1 + \frac{\alpha_p}{\beta_p - 1}\right) \tag{4.33}$$

with variance

$$Var\left(\sum_{k=0}^{n} \tau_k\right) = n\left(1 + \frac{\alpha_p}{\beta_p - 2}\right). \tag{4.34}$$

This implies that the estimated number of ensemble members up to time T is $\widehat{\ell_T} = \left\lceil T\frac{\beta_p - 1}{\alpha_p + \beta_p - 1} \right\rceil$.

2. *If the interarrival times τ_j correspond to a given T and λ generated from Beta (β_p, α_p), then*

$$E(\ell_T) = \frac{\beta_p}{\alpha_p + \beta_p} T. \tag{4.35}$$

3. *If each component of pair (λ_j, τ_j) are independent copies of λ and τ_j corresponds to λ_j, then*

$$E(\ell_T) = T\frac{\beta_p - 1}{\alpha_p + \beta_p - 1}. \tag{4.36}$$

In both cases 2) and 3), ℓ_T is distributed as Poisson with parameter $T/E\tau_1$, which implies that $Var(\ell_T) = E(\ell_T)$ and the estimation of ℓ_T is

$$\widehat{\ell_T} = T/\overline{\tau}. \tag{4.37}$$

Proof. We show only the first statement; the rest of the lemma is well known. If $\lambda \in (0,1)$ is distributed as *Beta* (α_p, β_p), then $1-\lambda$ is distributed as *beta* (β_p, α_p). The expected value of time is calculated in two steps; first, we take the conditional expectation, namely,

$$\begin{aligned} E\tau = EE(\tau|\lambda) &= \int_0^1 \int_0^\infty t\lambda \exp(-\lambda t)\, dt b(\lambda; \beta_p, \alpha_p)\, d\lambda \\ &= \frac{\Gamma(\beta_p - 1)}{\Gamma(\alpha_p + \beta_p - 1)} \frac{\Gamma(\alpha_p + \beta_p)}{\Gamma(\beta_p)} = 1 + \frac{\alpha_p}{\beta_p - 1}, \end{aligned} \tag{4.38}$$

where we assumed that $1 < \beta_p < \alpha_p$. Suppose $2 < \beta_p < \alpha_p$ to calculate the variance in a similar manner

$$Var(\tau) = EE\left((\tau - E(\tau|\lambda))^2 \middle| \lambda\right) = \int_0^1 \frac{1}{\lambda^2} b(\lambda; \beta_p, \alpha_p)\, d\lambda = 1 + \frac{\alpha_p}{\beta_p - 2}. \tag{4.39}$$

□

4.9 Predicted epidemic curves for COVID-19 in different countries

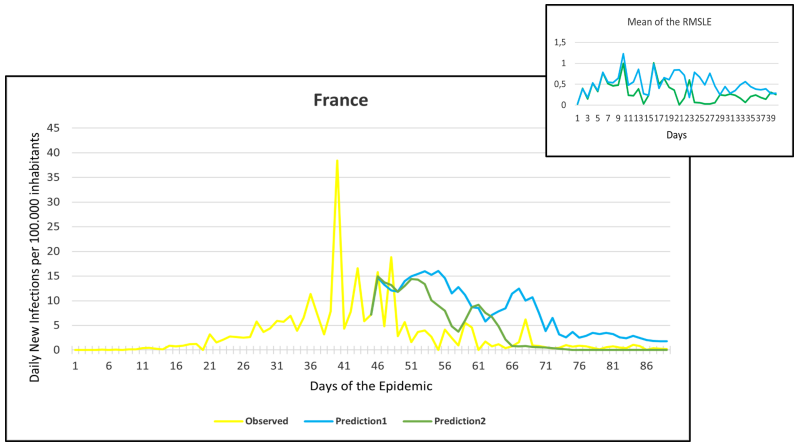


Figure 4.6: Observation and predictions for France During the First Pandemic Wave. The small graph in the upper right corner shows the daily error values calculated for the predictions

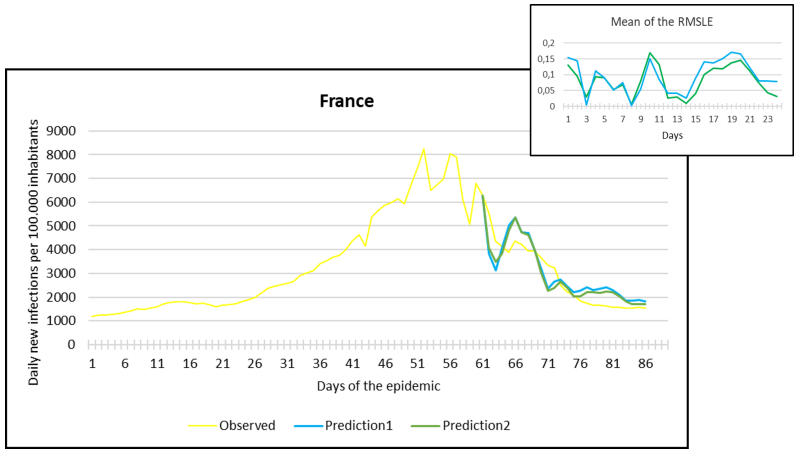


Figure 4.7: Observation and predictions for France During the Second Pandemic Wave. The small graph in the upper right corner shows the daily error values calculated for the predictions

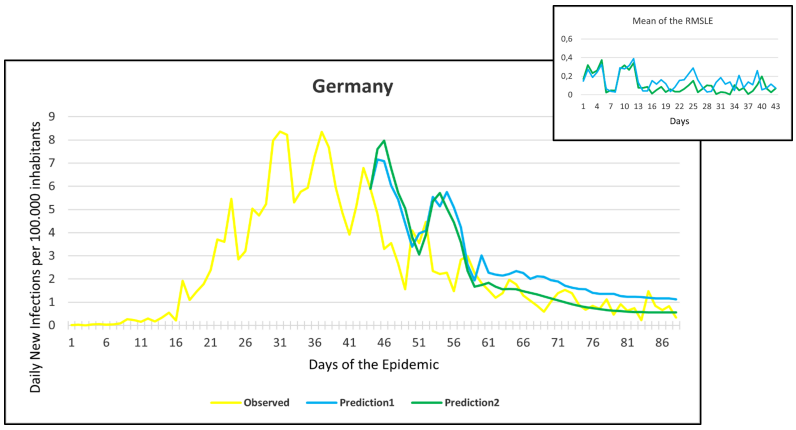


Figure 4.8: Observation and predictions for Germany During the First Pandemic Wave. The small graph in the upper right corner shows the daily error values calculated for the predictions.

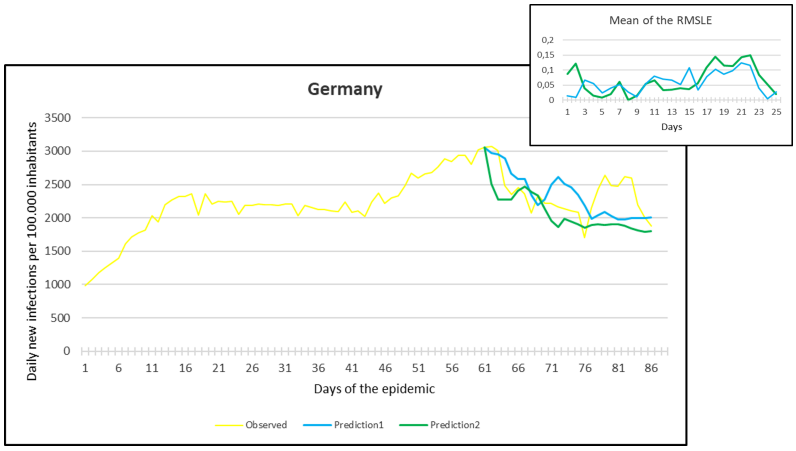


Figure 4.9: Observation and predictions for Germany During the Second Pandemic Wave. The small graph in the upper right corner shows the daily error values calculated for the predictions.

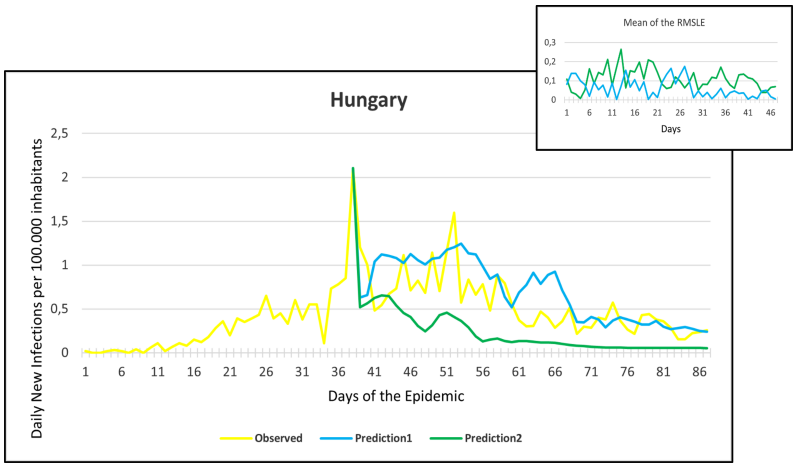


Figure 4.10: Observation and predictions for Hungary During the First Pandemic Wave. The small graph in the upper right corner shows the daily error values calculated for the predictions.

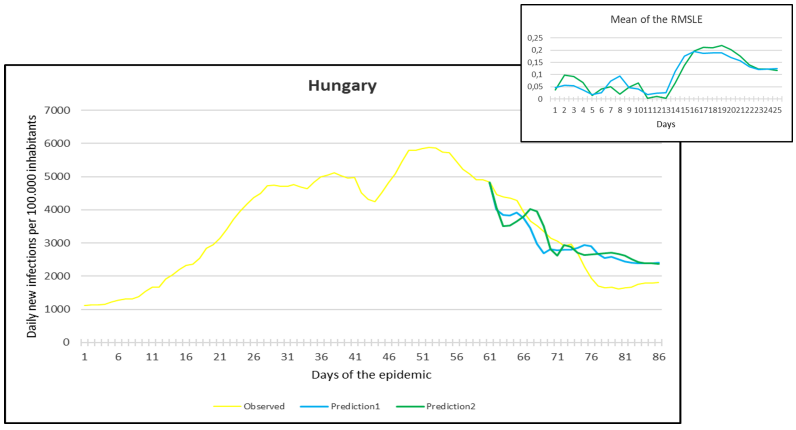


Figure 4.11: Observation and predictions for Hungary During the Second Pandemic Wave. The small graph in the upper right corner shows the daily error values calculated for the predictions.

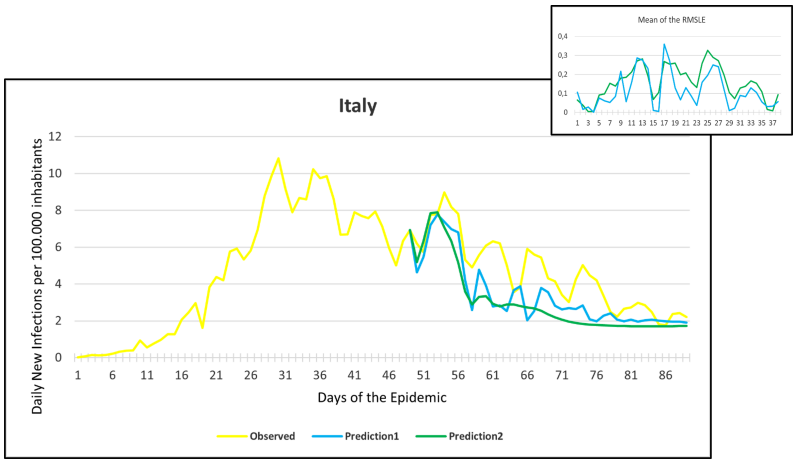


Figure 4.12: Observation and predictions for Italy During the First Pandemic Wave. The small graph in the upper right corner shows the daily error values calculated for the predictions.

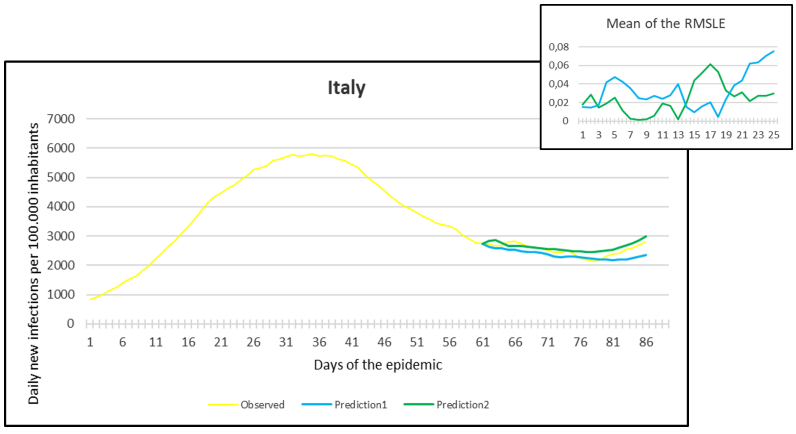


Figure 4.13: Observation and predictions for Italy During the Second Pandemic Wave. The small graph in the upper right corner shows the daily error values calculated for the predictions.

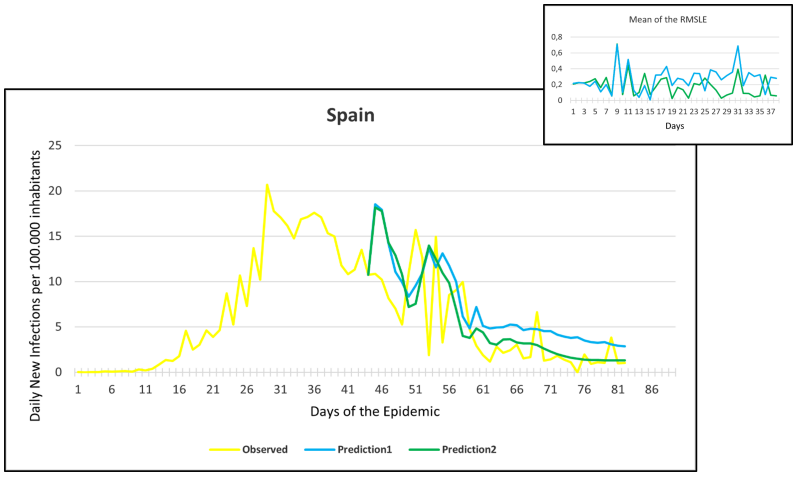


Figure 4.14: Observation and predictions for Spain During the First Pandemic Wave. The small graph in the upper right corner shows the daily error values calculated for the predictions.

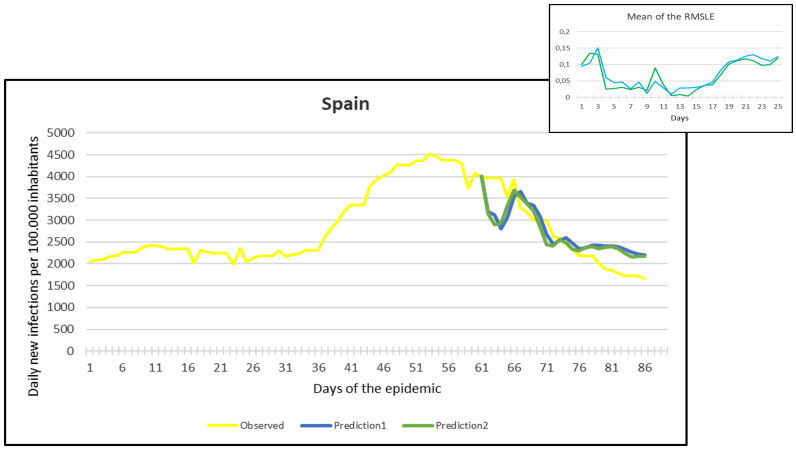


Figure 4.15: Observation and predictions for Spain During the Second Pandemic Wave. The small graph in the upper right corner shows the daily error values calculated for the predictions.

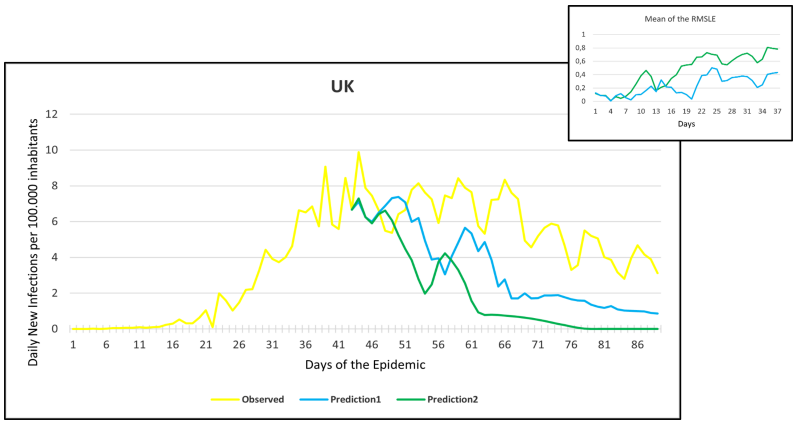


Figure 4.16: Observation and predictions for UK During the First Pandemic Wave. The small graph in the upper right corner shows the daily error values calculated for the predictions.

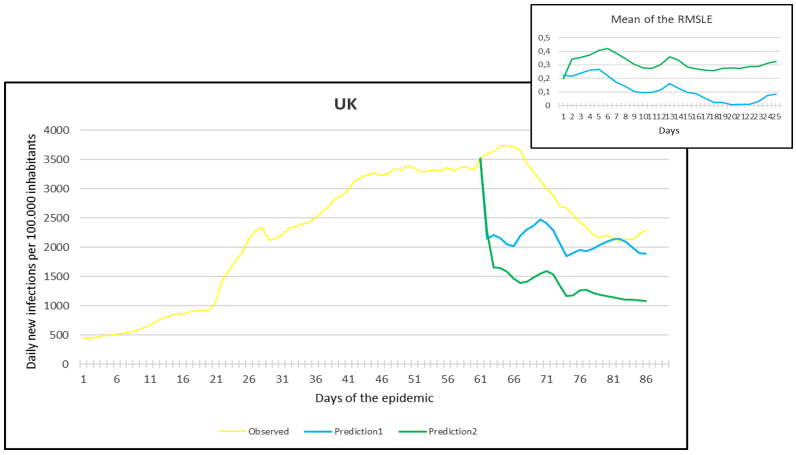


Figure 4.17: Observation and predictions for UK During the First Pandemic Wave. The small graph in the upper right corner shows the daily error values calculated for the predictions.

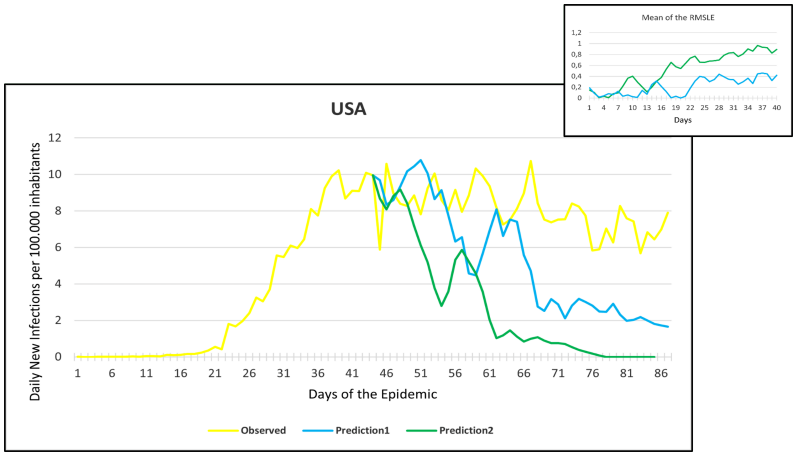


Figure 4.18: Observation and predictions for USA During the Second Pandemic Wave. The small graph in the upper right corner shows the daily error values calculated for the predictions.

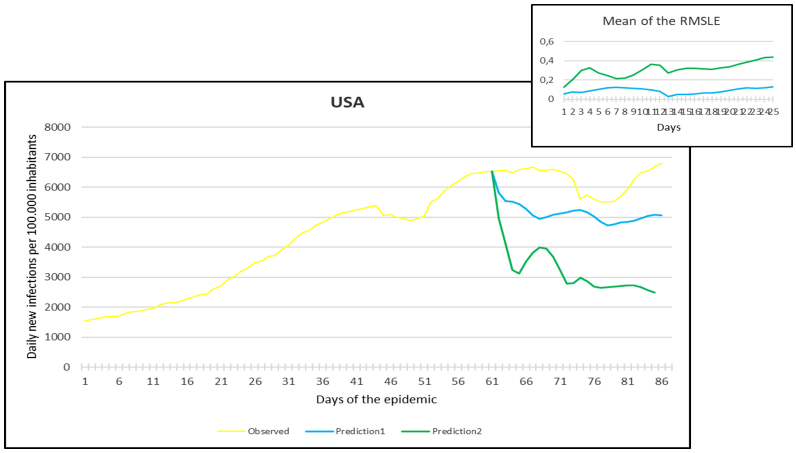


Figure 4.19: Observation and predictions for USA During the Second Pandemic Wave. The small graph in the upper right corner shows the daily error values calculated for the predictions.

4.10 Comparing search strategies on the UCI test subsets

Table 4.2: Comparing the proposed search strategy (SHErLoCk) with other selection methods on binary classification problems of the UCI test subsets using an ensemble pool of $n = 30$ classifiers.

		Dataset (size)	MAGIC (19 020)	Spambase (4 601)	HIGGS (20 000)	EEG (14 980)	Musk (6 598)	Breast (699)	Mushroom (8124)	Gisette (13 500)	Adult (48 842)	Average
		Method										
Ensemble accuracy	MAXSTEP	SHErLoCk	93.11%	**96.81%**	75.33%	96.02%	98.19%	**98.37%**	**99.52%**	**93.22%**	**88.03%**	**93.18%**
		SA+	**93.56%**	95.98%	75.03%	96.02%	98.23%	96.91%	99.17%	92.72%	86.19%	92.65%
		SA	93.08%	96.09%	74.33%	95.99%	98.23%	96.90%	98.36%	**93.22%**	86.92%	92.57%
		Genetic+	92.97%	94.23%	75.39%	96.12%	97.19%	96.63%	99.03%	92.01%	87.51%	92.34%
		Genetic	93.08%	95.88%	**75.77%**	**96.35%**	**98.48%**	95.73%	99.09%	92.49%	87.18%	92.67%
		Pruning	92.03%	96.21%	74.68%	95.01%	98.26%	96.08%	98.79%	91.06%	86.72%	92.09%
	STOP	SHErLoCk	92.47%	95.04%	**75.41%**	94.04%	**99.33%**	96.81%	98.47%	**92.24%**	**86.88%**	92.30%
		SA+	92.43%	**95.51%**	75.14%	94.43%	98.59%	96.45%	**99.18%**	**92.24%**	**86.88%**	**92.32%**
		SA	92.35%	94.82%	74.39%	93.78%	99.04%	**96.84%**	98.00%	92.17%	86.64%	92.00%
		Genetic+	92.41%	94.76%	74.89%	**95.16%**	98.98%	95.74%	97.71%	92.07%	86.81%	92.05%
		Genetic	**92.84%**	95.22%	73.70%	94.53%	99.03%	96.43%	97.99%	91.95%	85.93%	91.95%
		Pruning	91.88%	94.76%	72.91%	95.39%	97.56%	96.08%	97.51%	92.17%	86.42%	91.63%
	DET	Forward	88.11%	94.78%	72.22%	94.17%	97.08%	95.12%	98.54%	90.21%	82.27%	90.27%
		Backward	88.87%	94.78%	71.52%	94.17%	97.08%	95.07%	98.54%	90.21%	81.49%	90.19%
Computational time (secs)	MAXSTEP	SHErLoCk	4 **6.47**	**37.90**	**49.46**	**30.20**	**48.89**	**32.19**	**37.06**	**51.45**	**34.73**	**40.92**
		SA+	121.9	100.98	128.90	145.57	108.21	101.73	106.13	132.43	96.51	115.82
		SA	79.06	80.12	93.47	71.39	67.26	77.25	60.19	95.98	72.28	77.44
		Genetic+	63.42	68.03	59.57	61.07	74.02	69.09	89.67	67.03	69.08	69.00
		Genetic	46.08	48.92	53.22	58.16	51.23	48.56	57.99	64.42	49.18	53.08
		Pruning	125.84	107.08	115.42	129.45	93.06	90.98	103.54	137.08	151.08	117.06
	STOP	SHErLoCk	**0.98**	**0.39**	**0.93**	**0.38**	**0.92**	**0.74**	**0.39**	**0.43**	**0.28**	**0.60**
		SA+	8.41	4.67	6.73	8.47	7.33	5.09	8.79	5.88	6.11	6.83
		SA	8.78	6.92	4.56	7.93	8.59	9.48	6.95	5.89	7.99	7.45
		Genetic+	5.53	5.96	6.45	9.97	5.15	7.56	6.93	7.91	5.69	6.79
		Genetic	6.78	6.23	6.49	10.98	8.52	10.43	6.02	5.62	8.43	7.72
		Pruning	13.61	12.21	14.91	18.85	11.07	12.98	14.03	13.57	12.09	13.70
	DET	Forward	0.28	0.19	0.41	0.29	0.28	0.38	0.46	0.49	0.73	0.39
		Backward	0.21	0.37	0.43	0.42	0.31	0.84	0.35	0.25	0.23	0.38

Table 4.3: Comparing the proposed search strategy (SHErLoCk) with other selection methods on binary classification problems of UCI test subsets using an ensemble pool of $n = 100$ classifiers.

		Dataset (size)	MAGIC (19 020)	Spambase (4 601)	HIGGS (20 000)	EEG (14 980)	Musk (6 598)	Breast (699)	Mushroom (8124)	Gisette (13 500)	Adult (48 842)	Average
		Method										
Ensemble accuracy	MAXSTEP	SHErLoCk	95.06%	**97.18%**	**76.66%**	95.85%	99.00%	**99.06%**	**99.98%**	**93.86%**	87.94%	**93.84%**
		SA+	94.93%	96.45%	76.08%	96.01%	99.08%	98.13%	99.96%	93.22%	87.29%	93.46%
		SA	95.09%	96.89%	76.13%	95.91%	98.83%	98.81%	99.64%	92.88%	87.04%	93.47%
		Genetic+	95.09%	96.49%	76.48%	96.15%	99.07%	98.87%	99.80%	93.28%	**88.29%**	93.72%
		Genetic	**95.91%**	97.02%	76.27%	**96.39%**	**99.16%**	98.51%	99.64%	92.97%	87.93%	93.76%
		Pruning	94.76%	95.99%	75.39%	95.34%	98.95%	98.09%	99.27%	92.66%	86.95%	93.04%
	STOP	SHErLoCk	**94.27%**	95.71%	75.89%	94.79%	99.42%	**98.25%**	**99.63%**	93.02%	**87.18%**	**93.13%**
		SA+	93.91%	95.74%	**76.27%**	95.19%	99.58%	97.19%	99.17%	**93.21%**	86.61%	92.99%
		SA	94.11%	95.89%	**76.27%**	95.23%	**99.60%**	97.28%	99.52%	92.49%	86.77%	93.02%
		Genetic+	**94.27%**	95.59%	75.69%	95.08%	99.34%	97.93%	99.71%	93.05%	86.63%	93.03%
		Genetic	93.91%	**95.99%**	76.04%	**95.46%**	99.26%	98.08%	**99.63%**	92.93%	86.90%	**93.13%**
		Pruning	93.72%	95.80%	75.63%	94.07%	99.09%	97.82%	99.04%	92.71%	86.59%	92.71%
	DET	Forward	90.68%	95.02%	74.13%	93.01%	98.42%	95.61%	98.92%	90.92%	85.36%	91.34%
		Backward	90.82%	95.02%	74.19%	93.01%	98.42%	95.08%	98.91%	90.92%	85.83%	91.36%
Computational time	MAXSTEP	SHErLoCk	**194.12**	**206.89**	**191.91**	**203.88**	**214.28**	**201.03**	**186.67**	**159.32**	**178.01**	**192.90**
		SA+	349.62	290.89	390.82	278.56	253.59	375.71	313.56	301.25	311.87	318.43
		SA	251.02	291.13	269.22	278.39	269.59	228.44	286.23	259.92	258.62	265.84
		Genetic+	305.26	298.34	289.37	301.26	324.12	256.67	289.44	338.98	381.55	309.44
		Genetic	226.05	301.36	197.57	239.79	223.19	267.24	210.63	231.92	290.67	243.16
		Pruning	354.23	301.21	409.22	354.59	356.18	321.82	402.34	441.23	455.63	377.83
	STOP	SHErLoCk	**2.32**	**3.41**	**2.11**	**2.08**	**1.56**	**2.33**	**2.49**	**1.73**	**2.45**	**2.28**
		SA+	13.31	14.67	12.23	12.45	14.95	13.97	12.88	10.12	14.81	13.27
		SA	13.39	12.58	16.14	15.55	12.04	16.94	11.76	12.35	13.55	13.81
		Genetic+	12.56	12.61	15.95	13.37	13.49	16.68	14.88	13.14	12.59	13.92
		Genetic	13.53	13.58	12.88	15.47	11.51	13.67	12.02	11.27	12.28	12.91
		Pruning	19.41	11.78	17.32	36.36	21.69	31.55	22.92	15.34	19.53	21.77
	DET	Forward	0.59	0.72	0.81	0.92	0.69	0.89	0.93	0.96	0.94	0.83
		Backward	0.71	0.78	0.99	0.85	0.76	0.88	0.95	0.81	0.99	0.86