Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

Swiss Confederation

Federal Department of Home Affairs FDHA
Federal Office of Meteorology and Climatology MeteoSwiss

MeteoSwiss

Scientific Report MeteoSwiss No. 102

Automatic Gale Warning Proposals for Swiss Lakes and Regional Aerodromes

Lysiane Mayoraz, Jacques Ambühl

ISSN: 1422-1381

Scientific Report MeteoSwiss No. 102

Automatic Gale Warning Proposals for Swiss Lakes and Regional Aerodromes

Lysiane Mayoraz, Jacques Ambühl

Recommended citation:
Mayoraz L, Ambühl J : 2016, Automatic Gale Warning Proposals for Swiss Lakes and Regional Aerodromes, *Scientific Report MeteoSwiss,* **102,** 70 pp.

Editor:
Federal Office of Meteorology and Climatology, MeteoSwiss, © 2016

MeteoSwiss
Operation Center 1
CH-8058 Zürich-Flughafen
T +41 58 460 91 11
www.meteoswiss.ch

Abstract

Switzerland counts over 50 lakes and aerodromes for which warnings must be issued in case of high probability for wind gusts exceeding 25 knots. As these gale warnings are not automated, forecasters are required to permanently monitor the local weather evolution at these numerous specific locations, beside their main duties. Not surprisingly, quality assessments demonstrate low efficiency, with storm events frequently missed. In order to support the forecasters in their ongoing weather surveillance, an automatic warning system (GaleWarn) is developed which delivers precursory proposals for gale warnings. Working with a predictor list containing relevant measurements and short-term forecasts of the numerical weather prediction model COSMO-2, and later COSMO-1, GaleWarn delivers a storm probability for each warning object (lakes and aerodromes) every 10 minutes. Initially, a genetic programming method is developed and tested to forecast the storm probability. Nevertheless, the obtained results do not outperform the quality of logistic regression which is finally chosen for the operative system. GaleWarn has a high hit rate (75% on average) combined with a high false alarm ratio (68% on average). The performance is better in winter than in summer and the variation of quality among the warning objects is relatively large. Compared to the manual warnings, GaleWarn shows a significantly higher hit rate and a relatively higher false alarm ratio. Hence, the automatic warning proposals will help the forecaster to detect more storm events by reacting sufficiently early, but the numerous false alarms will have to be filtered. The warning proposals will appear on the platform the forecasters commonly use for the weather analysis and warning edition (NinJo). Several measures have been taken in order to reduce the number of warning proposals to the minimum required. We can deduce that, combined with the forecaster's expertise, the utilization of GaleWarn should lead to a general improvement of the gale warning quality.

Contents

1 Introduction

1.1 Context

Strong wind gusts represent a potential danger to aviation and navigation safety. Therefore, gale warnings must be issued for lakes and aerodromes when the probability of gusts exceeding 25 knots is high, so that affected people can take protective measures. At MeteoSwiss[1], gale warnings are issued for 51 lakes and aerodromes (map in Figure 1), a process which is not automatized at all. Consequently, forecasters are obliged to permanently monitor the local weather evolution at these numerous sites by looking at different data like pressure- and temperature-gradients (graphical interface of this empirical method in Figure 2). This task, which requires much attention to be performed correctly, must be conducted beside many other forecasting duties. Not surprisingly, quality assessment demonstrates that the efficiency concerning this type of warning is low with storm events frequently missed (Ambühl, 2008).

A system which can support the forecasters in their ongoing weather surveillance would thus be of great value. In this context, an evolutionary algorithm established in the field of genetic programming has been developed in collaboration with the Artificial Intelligence Laboratory of the University of Zurich in the framework of the two Bachelor theses Züger (2010) and Voisard (2010). Züger (2010) investigated the suitability of the algorithm to predict strong wind gusts. The promising results encouraged us to use this method for the subproject presented here called GaleWarn.

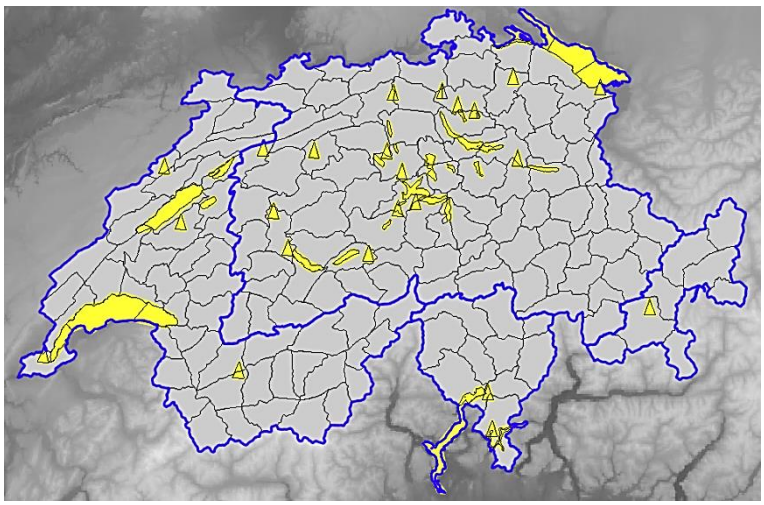


Figure 1: Swiss lakes and aerodromes for which gale warnings are issued (yellow).

1 Swiss Federal Office of Meteorology and Climatology.

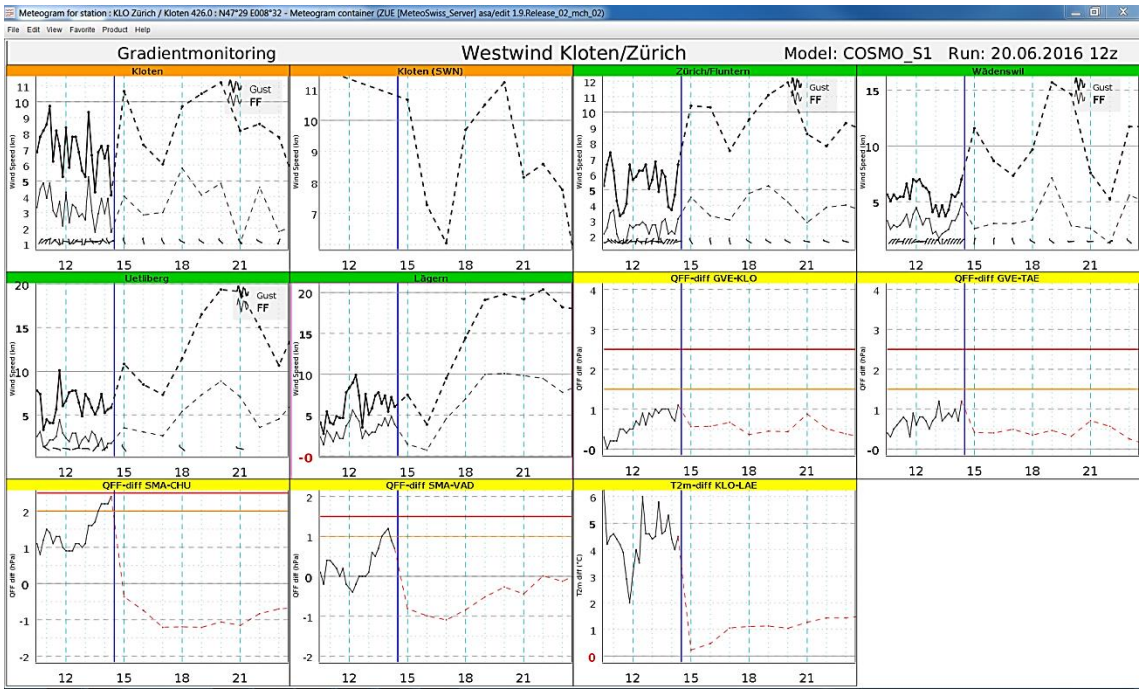


Figure 2: Empirical method used by the weather forecasters to decide if a gale warning should be sent for a specific lake or aerodrome. The diagrams show the temporal evolution of wind measurements and forecasts (direct model output from the numerical weather prediction model COSMO-1) at neighboring stations, as well as pressure and temperature gradients measurements and forecasts between relevant stations. The horizontal yellow and red lines represent empirical thresholds above which the weather forecasters mostly issue a warning.

1.2 Goals

GaleWarn is part of the broad OWARNA Umsetzung project, where weather warnings at MeteoSwiss are aimed to be optimized and modernized (Kube, et al., 2016). GaleWarn deals exclusively with the gale warnings presented above and has the following goals:

- To develop and implement an automatic gale warning system which delivers precursory warning proposals in advance when a potential for strong wind gusts appears, so that the forecaster can keep alert and send a warning in time, before the first wind gust over 25 kt occurs.

- To adjust and improve the genetic programming method of Voisard (2010) and Züger (2010) delivering probabilistic gale forecasts and evaluate the benefit of this method compared to a more standard statistical method (logistic regression).

- To operationalise and integrate the warning system on the platform the forecasters commonly use for the weather analysis and warning edition (NinJo), so that the processing of warning proposals is fully integrated in their daily tasks and can be performed properly and efficiently.

2 Methods

2.1 Overview

An overview of the processes conducted in the GaleWarn system is sketched in Figure 3. Two main activities have to be distinguished: the learning phase and the operational routine. In the former, a list of parameters consisting of past observation data and COSMO-2[2] short-term forecasts is chosen specifically for each warning object. These wind-relevant parameters which build the so-called predictors list are then used for the model training. The outcome of the training is a model which combines the parameters of the predictors list in order to deliver the probability of strong wind gusts in the next hours. A verification of this probability forecast is finally conducted in order to determine the optimal probability threshold q* beyond which an alarm proposal should be sent. All steps mentioned above constitute the learning phase which has to be conducted once for each warning object.

In the operational routine, the predictor list used in the learning phase consists of the latest data available. Based on it, the model delivers the current probability P that a wind gust over 25 kt occurs in the next hours at a specific warning object. In a last step, the obtained probability P is compared with the optimal probability threshold q* determined during the learning phase: if P is larger or equal to q*, an alarm proposal is delivered.

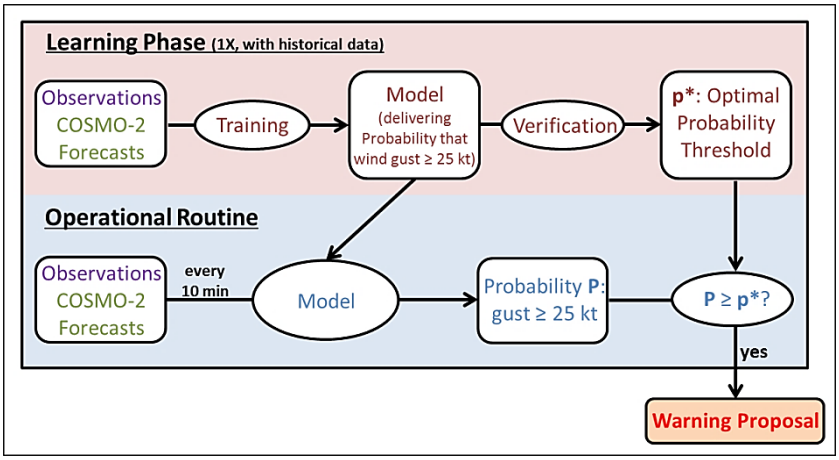


Figure 3: Overview of the processes conducted in the GaleWarn system. The learning phase that is conducted once for each warning object is shown in red, the operational routine in blue.

2 COSMO-2: 2-km numerical weather prediction model run by MeteoSwiss (COSMO-Prognosesystem, 2015).

In the next subchapters, the different steps of the learning phase are explained in more detail. An emphasis is put on the genetic programming method used initially to deliver the probabilistic forecast, since a large part of the project consisted of the adjustments and ameliorations of this technique. In section 2.3.2, the logistic regression method developed by Zimmermann (2014) to get also probabilistic forecasts for wind gusts is only described briefly because no method adjustments or ameliorations were accomplished within the frame of this project. The method is simply applied as such on the GaleWarn predictors list in order to get a benchmark and evaluate the genetic programming method.

2.2 Predictor list

Figure 4 shows a schematic view of the process repeated every 10 minutes. The time is represented on the horizontal axis and the red star corresponds to the current time t_0. At time t_0, the forecaster wonders if a gust over 25 kt will occur in the time period $t_{0+30min}$ to t_{0+3h} (time interval corresponding to the definition of a hit, which is a warning sent at least 30 minutes and at most 3 hours before the first wind gust over 25 kt (Schraner & Reynolds, 2015)). In parallel, at time t_0, the GaleWarn system takes into account observational data of the last hour and COSMO-2 forecasts for the next 1 to 3 hours that could be relevant for the wind gust prediction. These parameters constitute the predictors that will be used to predict the maximal wind gust at the representative stations in the time interval $t_{0+30min}$ to t_{0+3h} (the predictand).

The measurement data are essentially delivered from SMN-stations[3], but also from SMART-stations[4]. For a specific location, measurement data and COSMO-2 forecasts are taken at the representative stations as well as at neighbouring and altitude stations (situated at higher altitude). Based on these data, several predictors are calculated, forming the predictor list. Here are some examples of predictors: temperature difference over time at a specific station, pressure gradient between two distinct stations, maximal CAPE[5] predicted for the next hours at a particular station, etc. A list of the representative stations and standard predictors used for every warning object was defined by an experienced weather forecaster and is listed in the Appendix A. The stations used for the calculation of the standard predictors vary for each warning object. Some additional predictors were also taken into account for some specific warning objects.

3 SwissMetNet: automatic observational network from MeteoSwiss (Automatisches Messnetz, 2014).

4 Observational network of MeteoSwiss for the aviation weather (Beobachtungsnetz für Flugwetter, 2014).

5 Convective available potential energy.

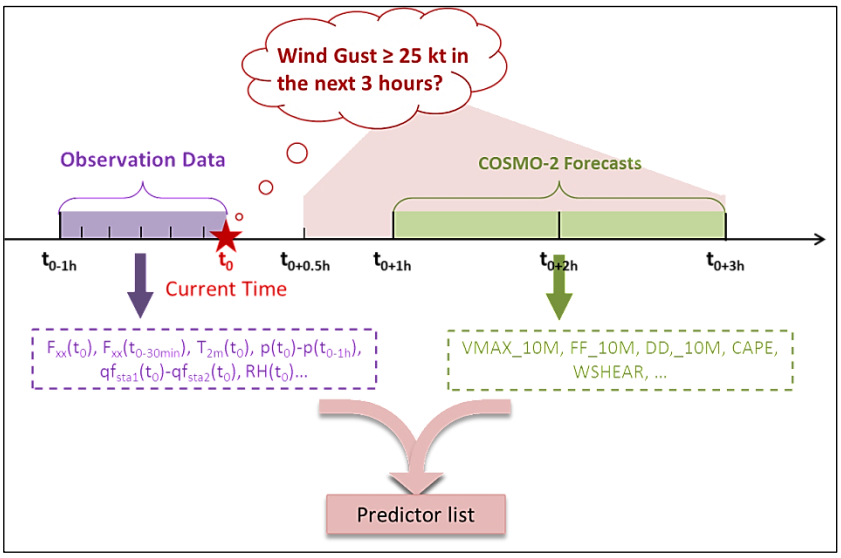


Figure 4: Temporal situation of the object warning problematic. The horizontal axis represents the time and the red star t_0 the current time step.

Figure 5 presents a schematic view of the predictor list's setup, subdivisions and functionalities. The predictors and the corresponding predictand are calculated based on a set of 4-year past data with a resolution of 10 minutes. The calculation is not applied to every time step of the 4-year data set, but only to the two relevant categories of time steps fulfilling the following conditions:

1. No wind gust over 25 kt measured at the representative stations from t_{0-3h} to $t_{0+30min}$, but wind gust over 25 kt measured at the representative stations between $t_{0+30min}$ and t_{0+3h}.
   ➔ pre-storm cases

2. No wind gust over 25 kt measured at the representative stations from t_{0-3h} to t_{0+3h}.
   ➔ no-storm cases

The first category corresponds to the time steps at which a storm warning should be sent to be considered as a hit, and not as a miss or a false alarm: there has been no wind gust over 25 kt in the last 3 hours (the time step is not part of an already existing storm event) but there will be a wind gust over 25 kt in the time period $t_{0+30min}$ and t_{0+3h}. The second category represents the time steps not belonging to any storm event (no gust over 25 kt measured 3 hours before, till 3 hours after) so that no warning should exist at these time steps, otherwise it would be considered as a false alarm. Note that only the relevant information about the beginning of a storm event is considered, given the goal of GaleWarn to alert the forecaster before a strong wind gust has occurred. The tool is not trained to determine when a storm event finishes but when it starts.

The predictors and predictand calculation is then completed on these two time steps categories and the data set built is called predictor list. The pre-storm cases represents only approximately 2% of the predictor list, the no-storm cases about 98%. The predictor list is split into two parts: the first 2 years of data represent an independent verification dataset with time coherent time steps (orange data set in Figure 5) and the last 2 years of data is used for the learning process (green data set in Figure 5). In order to avoid overfitting during the training, the data used for the learning process is randomly

split into two parts forming the test- and learn-data on which training is conducted. The data amount contained in the learn- and test-data is reduced to 3000 time steps each, otherwise the evolutionary algorithm would work extremely slowly. The proportion of pre-storm cases in the learn- and test-data is artificially set to 25% in order to provide enough storm cases. Details on how this pre-storm proportion was chosen can be found in section 3.1.3.

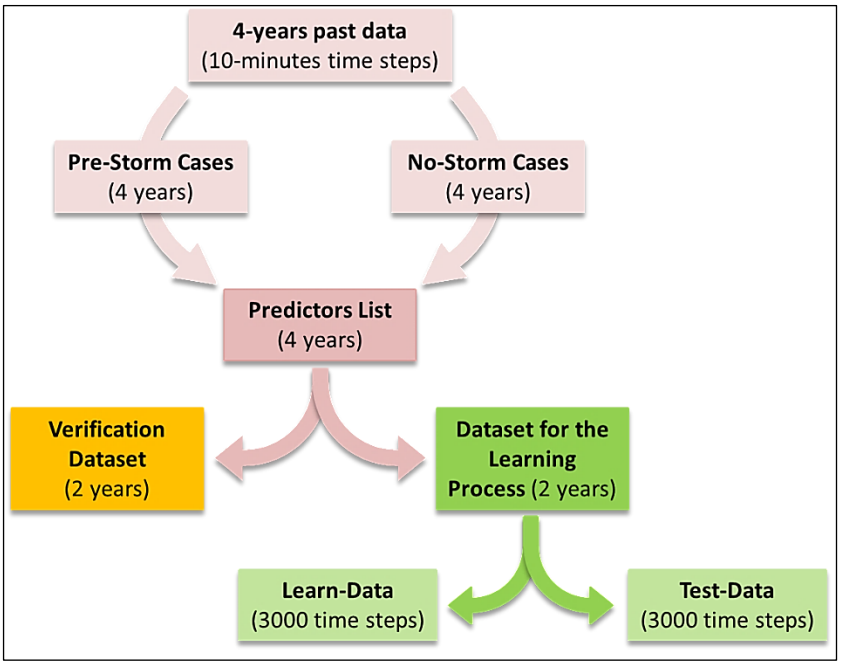


Figure 5: Scheme of the predictor list's building, subdivisions and functionalities.

2.3 Training

2.3.1 Genetic Programming (GP)

2.3.1.a) Overview

Genetic programming is the method initially chosen to get the best approximation of the next wind gusts from the predictor list. It is a type of evolutionary algorithm inspired by the evolution theory of species. This machine learning technique is used to optimize a population of computer programs in order to perform a given task (Floreano & Mattiussi, 2008). In our case, this task is the forecast of the maximal wind gust between $t_{0+30min}$ and t_{0+3h} at a specific lake or aerodrome. The main advantage of this method is the high complexity the obtained solutions can have, which gives the chance to the model to describe the complex and various phenomena leading to strong wind gusts (*IfThenElse* statements for example).

During the genetic evolution, the following steps are conducted (Poli, Langdon, McPhee, & Koza, 2008) (graphic representation in Figure 6):

1. A random population of computer programs is created based on the predictor list, constituting the generation 0 of computer programs. The population number is set to 1000 programs.
2. Each program is evaluated with the help of a fitness function in order to determine how successful the programs are in predicting the next wind gusts.
3. The best programs are selected and genetic operations like crossing and mutation are applied to the selected programs in order to build a new generation of programs.
4. Finally, steps 2 and 3 are repeated until the maximum number of 200 generations is reached.

At the end of the evolution, the program with the highest fitness value and fulfilling the overfitting test explained in section 2.3.1.d) is chosen.

The whole process is then repeated 20 times, in order to get a set of 20 different programs, in our case Java methods, delivering a probability distribution function from which a probability of exceedance can be computed. Indeed, the set of Java methods provides an ensemble forecast system for the maximal wind gust in the next hours.

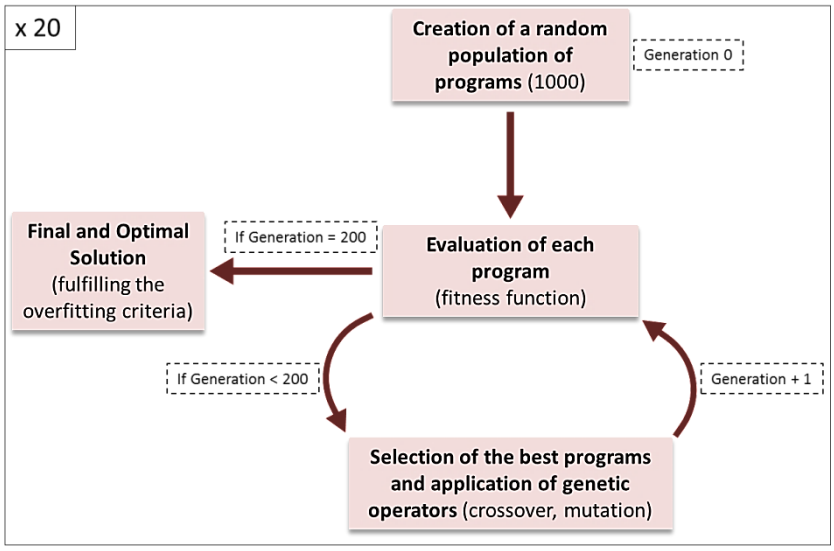


Figure 6: Schematic view of the genetic programming method. The whole process is repeated 20 times in order to get 20 different programs building in operation an ensemble forecast of 20 members.

2.3.1.b) Performance Evaluation

The fitness function used to evaluate the performance of each program is defined as follows:

$$Fitness = Hit\ Rate * (100\% - False\ Alarm\ Ratio)$$

The *Hit Rate (HR)* and *False Alarm Ratio (FAR)* are in % and defined as follows (Wilks, 2006):

$$HR = Hit / (Hit + Miss)$$

$$FAR = False / (False + Hit)$$

Where *Hit* : Number of correct alarms
Miss : Number of missed alarms
False : Number of false alarms

For the performance evaluation of the programs, the wind gust prediction of each time step contained in the learn- and test-data is individually looked at. A hit is counted if the program predicts a gust over 25 kt that actually occurs in the time period $t_{0+30min}$ and t_{0+3h}. A false alarm is counted if the program predicts a gust over 25 kt that actually does not occur in the time period $t_{0+30min}$ and t_{0+3h}, and a miss is counted if the program does not predict a gust over 25 kt which actually occurs in the time period $t_{0+30min}$ and t_{0+3h}. At the end, the total number of hits, misses and false alarms are summed up and the scores HR and FAR calculated to get the value of the fitness function, called also “fitness”.

2.3.1.c) Genetic Operations

During the genetic evolution, the solutions selected are mutated and crossed to build the next generation of computer programs. These mutation and crossover processes are called genetic operations and are similar to the processes happening in genetics in the natural world.

In order to better illustrate how these genetic operations actually work, the computer programs obtained during the evolution are represented as tree structures. Figure 7 gives the example of the simple algorithm max(x+x, x+3*y). The functions max, plus and times are the nodes and the leafs are the variables x, 3 and y. The depth of the algorithm corresponds to the number of nodes that have to be crossed to reach the deepest leafs (Poli, Langdon, McPhee, & Koza, 2008). In the case of Figure 7, the algorithm depth is equal to 3.

The left diagram of Figure 8 illustrates a crossover operation where randomly selected parts of two parent programs are combined to build the offspring program. The right diagram shows a mutation operation where a randomly chosen part of a parent program is modified to create the offspring program (Poli, Langdon, McPhee, & Koza, 2008).

The evolutionary algorithm used in this project was developed during the Bachelor theses Voisard (2010) and Züger (2010) and is based on the open source Java genetic algorithm package JGAP (Meffert & Rotstan, 2015).

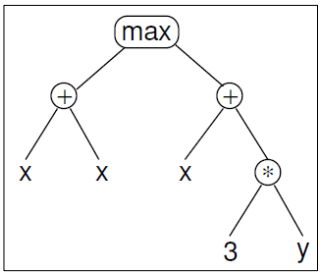


Figure 7: Tree representation of the mathematical function max(x+x, x+3*y). (Poli, Langdon, McPhee, & Koza, 2008)

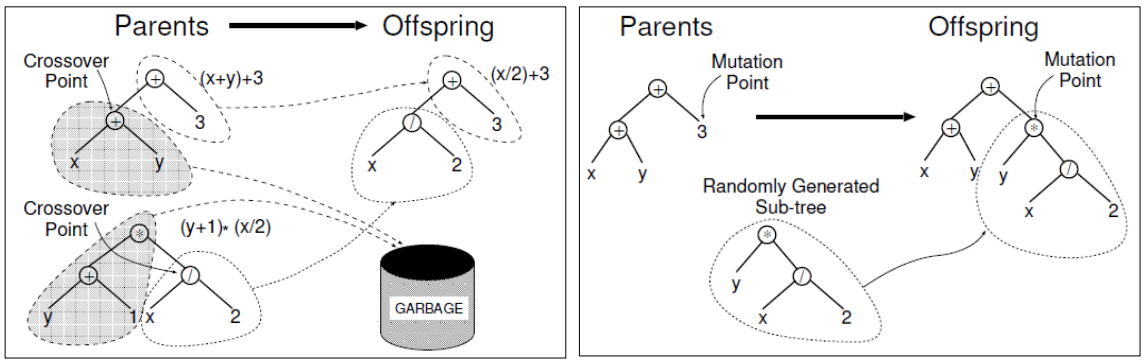


Figure 8: Left: Illustration of a crossover operation. The two parents algorithms (x+y)+3 and (y+1)*(x/2) are represented as a tree structure. A subtree is taken away from each parents algorithm's copy and the remaining subtrees are put together to form the offspring algorithm (x/2) +3. Right: Illustration of a mutation operation. The parent algorithm (x+y)+3 is represented as a tree structure. The leaf corresponding to the number 3 is replaced by the randomly generated sub-tree y*(x/2) to form the offspring algorithm (x+y)+(y*(x/2)). (Poli, Langdon, McPhee, & Koza, 2008)

2.3.1.d) Overfitting Test

After a certain number of generations, genetic programming has the tendency to deliver very complex solutions that can predict the training data almost perfectly but that show poor performance when used on an independent data set. This undesired phenomenon is called overfitting (Wilks, 2006). In order to avoid this, two independent data sets are used during the evolution. The learn-data is the data set with which the evolutionary algorithm is trained. After each new generation, the solutions obtained are evaluated by calculating the fitness function on the learn-data. The solutions showing the best fitness values are selected, mutated and recombined to create the next generation. At the same time, the fitness function of each solution is also calculated on the test-data. In this way, at the end of the evolution, the fitness values based on both data sets are tested and the optimal solution selected according to the following method:

1. Selection of the solution having the highest fitness value on the test-data (= Select)
2. If a solution exists fulfilling the following criteria, choose this one instead of Select:

   Criteria:

   a. The average of the learn- and test-data based fitness values is higher or equal to the fitness average of Select.

   AND

   b. The absolute difference between the learn- and test-data based fitness is smaller or equal to the fitness difference of Select.

3. If several solutions fulfil the criteria of point 2, select the one with the highest fitness average.

The implementation of this overfitting test provides the optimal solution found during a genetic evolution that has not been subjected to overfitting yet.

Figure 9 shows the evolution of the fitness function calculated with the learn- (in light red) and test-data (in dark red) during a full learning process. The average of both is sketched in green. The vertical axis represents the fitness value and the horizontal axis the new best solutions found after a generation (it is indeed not guaranteed that a better solution is found after each new generation: the variable depicted on the horizontal axis corresponds to the new best solutions and not to the generation number). In the first generations, both fitness values increase rapidly. A peak is then reached from where the fitness of the test-data starts to decrease whereas the fitness of the learn-data continues to increase. The solution selected by the overfitting test is represented by the vertical line situated in this case just before the new best solution number 40. It is relatively far in the evolution, but its test-data fitness is still high, meaning that no overfitting occurred at this stage yet.

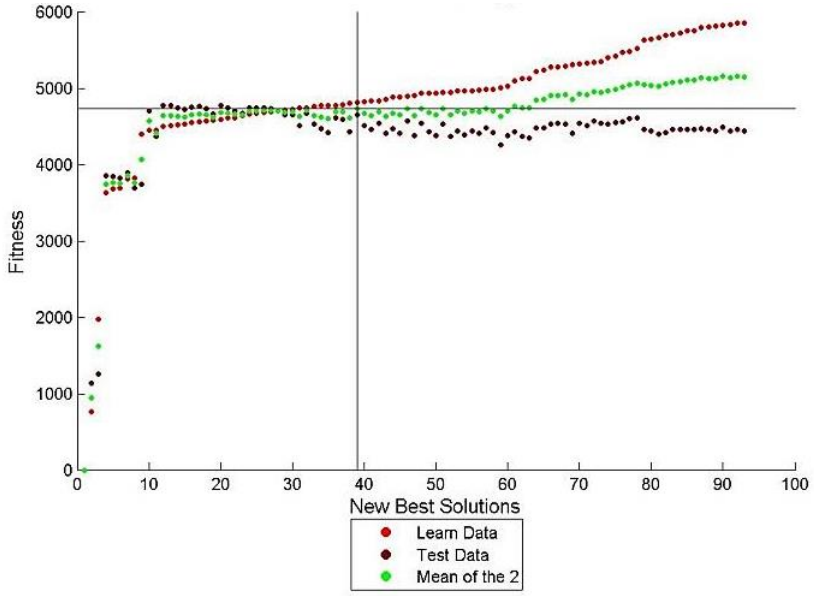


Figure 9: Evolution of the fitness value calculated with the learn-data (in light red) and with the test-data (in dark red) during a full learning process. The average of both fitness is sketched in green. The horizontal axis represents the new best solutions found after a generation and the vertical axis the corresponding fitness value. The vertical line corresponds to the solution selected according to the overfitting test.

2.3.1.e) Example of a Solution

Figure 10 depicts a typical Java method delivered by the genetic evolution. The mathematical functions like max, plus, minus etc. are written in orange, whereas the variables (each corresponding to a predictor) and constants are written in blue. The complexity of this solution becomes clearer when looking at the tree representation sketched at the bottom of Figure 10. The method has a depth of 12 and combines 7 distinct predictors: *ffma*, *mmo*, *dmo*, *qfdif*, *wshe*, *ttt* and *ff0* (see Appendix A for the predictors definition).

```
plus.evaluate (max.evaluate (mmo, ffma), sine.evaluate (minus.evaluate (dmo, max.evaluate (minus.evaluate
(pow.evaluate (min.evaluate (qfdif, sine.evaluate (1.71)), sine.evaluate (multiply.evaluate (4.16, mmo))),
pow.evaluate (min.evaluate (6.27, divide.evaluate (0.52, wshe)), log.evaluate (plus.evaluate (pow.evaluate
(min.evaluate (6.27, minus.evaluate (dmo, max.evaluate (fxxs, 6.12))), log.evaluate (plus.evaluate(ff0, ttt))),
```

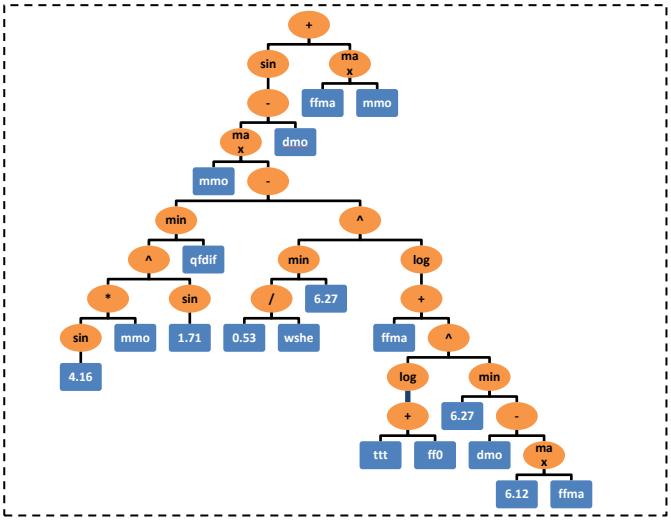


Figure 10: Example of a solution found after the genetic evolution. Top: Java method where the mathematical functions are written in orange, the variables and constants are written in blue. Bottom: Tree representation of the Java method. The nodes, containing the mathematical functions, are sketched in orange, whereas the leafs, containing the variables and constants, are written in blue.

2.3.2 Logistic Regression (LogReg)

In order to get a benchmark concerning the quality of the GP method for predicting wind gusts, another method was applied on the same set of predictors. The reference method used was developed by Zimmermann (2014) who applied a logistic regression on a set of COSMO-2 forecasts in order to predict probabilistic elements of TAF[6] including wind gusts.

Logistic regression is a statistical method estimating the relationship between a categorical variable and a set of independent variables x_i (predictors). It delivers the probability p that an event occurs (in our case, wind gusts ≥ 25 kt in the next hours) given the predictors x_i and the estimated regression coefficients β_i :

$$p = \frac{1}{1 + e^{-\sum_i x_i \beta_i}}$$

The logistic regression model is estimated on the learn data set (dark green data set in Figure 5). After an appropriate selection of the best predictors entering the regression equation (stepwise method based on likelihood ratio test (Hosmer, Lemeshow, & Sturdivant, 2013)), the associated regression coefficients are estimated by maximum likelihood. More details on this logistic regression method are contained in Zimmermann (2014).

6 Terminal Aerodrome Forecast: concise statement of the expected meteorological conditions at an airport during a specified period. (TAF Decoder, 2015)

2.4 Verification of the Probabilistic Forecast

Both methods described above provide a probabilistic forecast for wind gusts above 25 kt in the next hours. This forecast is verified in order to determine the optimal probability threshold above which an alarm proposal should be sent.

The verification occurs event-based, in contrast to the evaluation of the individual Java methods performed during the learning process of the GP method. As described in section 2.3.1.b), the latest evaluation is simply based on the single time steps contained in the learn- and test-data. So, several hits, misses, or false alarms belonging actually to the same warning event are summed up. This does not happen in the probabilistic forecast verification conducted after the learning process, because this verification considers the warning proposals and storm observations as events and not simply as individual time steps.

As already mentioned in section 2.2, the data set contains the so called pre-storm cases, time steps situated 30 minutes to 3 hours before a storm event begins and at which a warning should be sent in order to count as a hit, and the no-storm cases, time steps at which no warning should be issued. The verification on one hand consists in looking at the no-storm time steps at which the probabilistic forecast proposes nevertheless a warning, depending on probability thresholds between 5% and 100%. From these false alarm time steps, false alarm events are built, following the general rule that two warnings are considered as distinct events if there is a temporal interruption of more than 3 hours between them (= inter time). If the break between two warnings is shorter or equal to 3 hours, they belong to the same event (Schraner & Reynolds, 2015). On the other hand, the pre-storm cases are sorted into observed pre-storm events (following the same inter time rule), which are then compared to the warnings suggested by the probabilistic forecast at probability thresholds from 5% to 100%. If a warning is proposed at at least one time step of the observation event, it corresponds to a hit, whereas if no warning is proposed for an observed pre-storm event, it corresponds to a miss. The false alarms, hits and misses are then summed up depending on the probability threshold, and the scores Hit Rate (HR) and False Alarm Ratio (FAR) calculated (see formula in section 2.3.1.b)). This verification method follows the same principles as the automatic warning verification tool AWV1.0 basis verification (Weusthoff, 2012). Therefore, a comparison between the results of the AWV1.0 basis verification of the manual warnings and the results of the GaleWarn verification conducted here can be considered as fair.

Figure 11 shows an example of a pseudo ROC-curve[7], where the HR on the vertical axis and the FAR on the horizontal axis are represented as function of the probability threshold, written next to each point. The diagonal linking the top left corner (0%,100%) and the bottom right corner (100%,0%) represents the line where the bias[8] is equal to 1. Warning systems situated over the diagonal (bias > 1) induce an overforecasting of the storm events (Wilks, 2006). For example, if a

7 The term “pseudo” is used here because the curve represents the hit rate versus the false alarm **ratio**, whereas a ROC-curve (Relative Operating Characteristic) represents per definition the hit rate versus the false alarm **rate** (Wilks, 2006). In our case, the false alarm rate, representing the ratio of false alarms to the total number of non-storm events, does not deliver relevant information about the warning system because storms are so rare events, respectively the number of non-storm events is so high. The false alarm ratio is more adapted and hence represented on the horizontal axis.

8 The bias B represents the ratio of the number of forecasted events to the number of observed events (Wilks, 2006):
$B = (\mathrm{Hit} + \mathrm{False})/(\mathrm{Hit} + \mathrm{Miss}) = \mathrm{HR}/(1 - \mathrm{FAR})$
In the case of B = 1, HR = (1 – FAR)

system sends an alarm already at a probability of occurrence of 20%, the HR will be high, but the FAR will be high too. Note that an addressee choosing such a system would be risk adverse (Ambühl, 2010). On the other hand, warning systems situated under the diagonal (bias < 1) induce an underforecasting of the storm events (Wilks, 2006). For example, if a system sends an alarm only at a probability of occurrence of 80%, the FAR will be low, but the HR will be low too. In this case, an addressee choosing such a warning system would be risk friendly (Ambühl, 2010). An ideal warning system would hence be situated at the top left corner of the graph, with a HR of 100% and a FAR of 0%, and a bias equal to 1.

Based on this pseudo ROC-curve, the optimal probability threshold is determined, specifically for each warning object, as the one maximising the fitness and inducing a FAR of maximal 75% and if possible a HR of at least 75%. The fitness, which was already defined in section 2.3.1.b), corresponds to the green area sketched in Figure 11. Its maximal value is 10'000 (= 100%*100% [%^{2}]). The minimal HR of 75% is set as condition because of the primary goal of the tool which is to help the forecaster to detect more storm events in advance. However, as it will be detailed in chapter 3, the FAR induced by the automatic system is generally high. In order to limit it, the condition of a maximal 75% FAR was also fixed to set the optimal probability threshold. The limit values of 75% were set upon consultation with weather forecasters, the end-users of the tool.

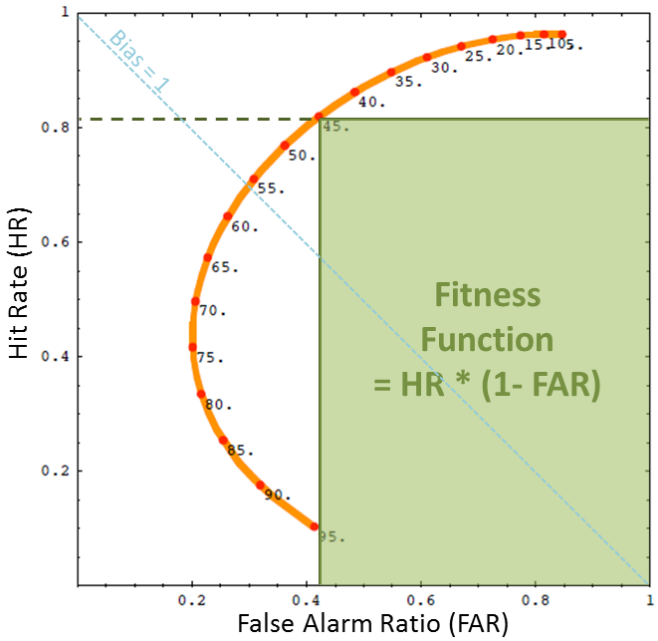


Figure 11: Example of a pseudo ROC-curve. The false alarm ratio (x-axis) and the hit rate (y-axis) are represented as function of the probability of occurrence above which a warning is sent (written next to each point). The fitness function HR*(100%-FAR) corresponds to the green area. Curve taken from Ambühl (2010).

3 Tests, Tuning and Results

3.1 Experiments with GP

In this section, the results of the main experiments conducted with GP during the development phase are presented. The experiments consisted in varying some training parameters and learn-data properties in order to analyse the influence of each parameter and find the optimal settings leading to the best possible results. The tests were mainly carried out on the warning object Grenchen (LSZG), and some of them were also conducted on the object Zürich-Kloten (LSZH). As the results of LSZH have been similar to the ones of LSZG, the obtained settings were thus generalised for all warning objects.

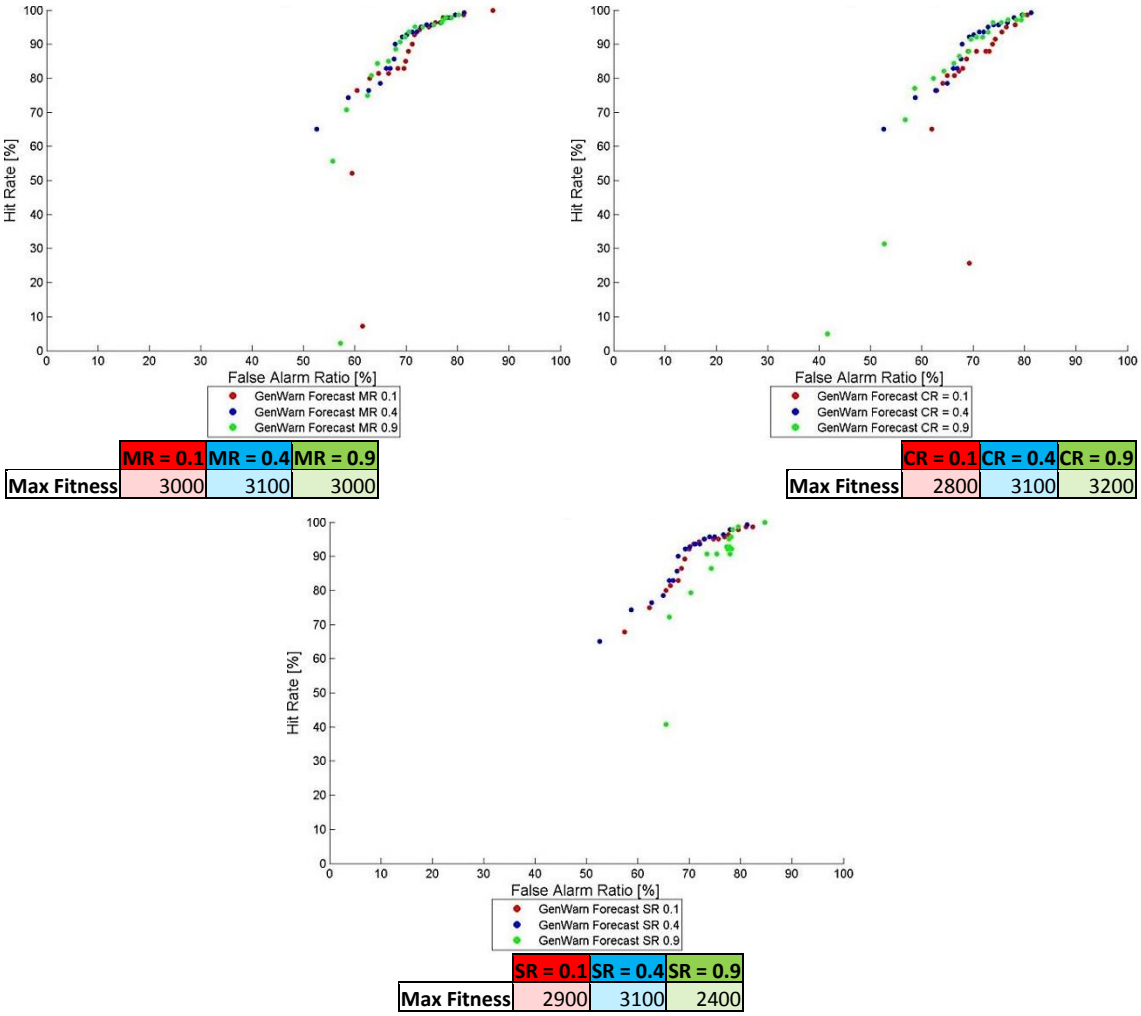
In the following sections, the results always refer to the aerodrome of Grenchen.

3.1.1 Variation of Genetic Programming Parameters

Some parameter settings have to be predefined to the genetic programming learning. They are listed in Table 1 which contains the definition of each specific term. In this section, the results of the sensitivity study of parameter settings on the forecast performance are presented.

The survival rate SR determines which proportion of the best solutions is selected to build the next generation. Mutation takes place only on a part of the selection, determined by the mutation rate MR, creating new solutions. Finally, crossing operations occurs on the existing solutions, at a rate defined by the crossover rate CR.

Each of these rates has been individually tested with the values 0.1, 0.4 and 0.9, while the other two rates were set constant (0.4). The performance of the solutions resulting from the different settings is represented as a pseudo ROC-curve with the best fitness values in Figure 12. A difference of fitness of 200 or more is considered as significant. This value has been assumed after looking at the variability of fitness of different learning processes conducted with exactly the same parameters. When looking at Figure 12, no significant difference is observed by varying the mutation rate, whereas a crossover rate of 0.1 shows a slightly lower performance compared to a higher crossover rate. A survival rate of 0.9, in contrast, shows a clearly lower performance as compared to lower ones. This observation can be explained by the fact that, if the survival rate is too high, the supply of new solutions after each generation is too low to enable an effective evolution towards good solutions. The influence of the mutation- and crossover rate seem on the other hand to be negligible in our case. After these experiences, the final setting of SR = 0.4, MR = 0.4 and CR = 0.4 has been chosen.



	MR = 0.1	MR = 0.4	MR = 0.9
Max Fitness	3000	3100	3000

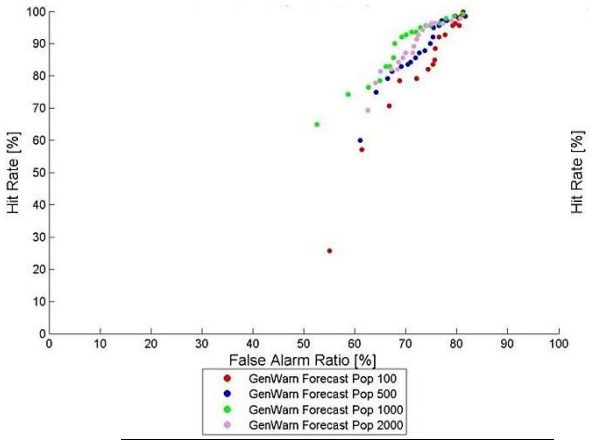
	CR = 0.1	CR = 0.4	CR = 0.9
Max Fitness	2800	3100	3200

	SR = 0.1	SR = 0.4	SR = 0.9
Max Fitness	2900	3100	2400

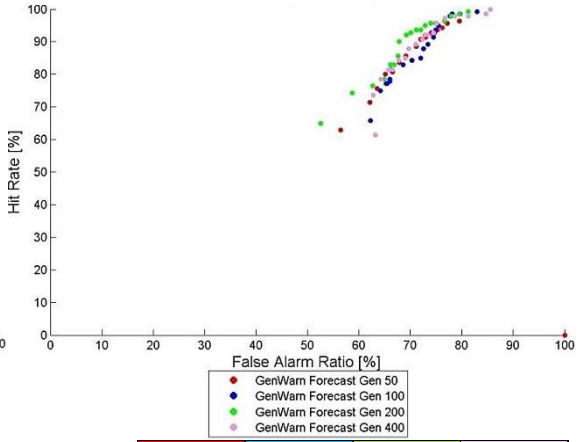
Figure 12: Verification conducted on the 2-year independent data set of the algorithms obtained for LSZG with different training parameters. On the top left, solutions with a varying mutation rate, on the top right, solutions with a varying crossover rate, and at the bottom, solutions with a varying survival rate. Each plot represents the FAR on the x-axis and HR on the y-axis as function of the probability of occurrence (0.1, 0.15, …, 0.95, 1, here not written next to each point because of the readability). Each colour corresponds to a specific setting (see the legend under the plots). The maximal fitness of each solution is reported in the table under each plot.

The population size (Pop) is the number of solutions contained in each generation. The generations number (Gen) determines the number of generations conducted before the evolution stops, and the maximal depth (Depth) defines the maximal complexity of a solution (see the exact definition of the depth in the paragraph 2.3.1.c)). These three parameters have a major influence on the speed and efficiency of the genetic evolution. With a too large population number, the evolution can take extremely long without showing any significant improvement after new generations. Combined with a large maximal depth, it can lead to the crash of the program because too much RAM is used. Therefore, a balance has to be found between the time and effort invested and the gain obtained (Voisard, 2013).

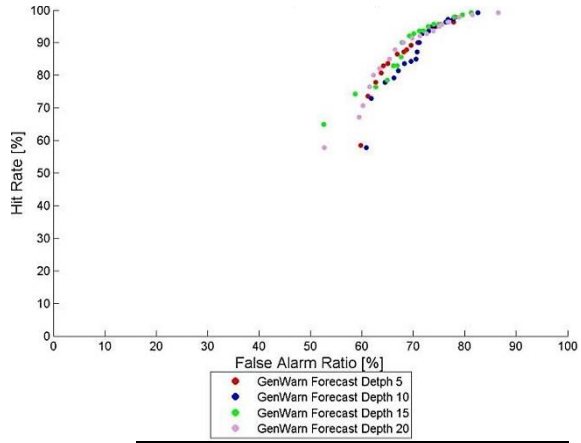
Figure 13 presents the comparison of performance between solutions that have been found with different parameter settings. On the top left, population sizes of 100, 500, 1000 and 2000 have been tested. A population size of 1000 seems to be optimal: lower population sizes show a clearly lower fitness and a population size of 2000, that implies a longer-lasting genetic evolution, does not show a better performance than a population size of 1000, on the contrary. On the top right, the generation numbers of 50, 100, 200 and 400 have been compared. A generation number of 200 shows the maximal fitness while the evolution duration remains tolerable (about 30 hours). At the bottom, depths of 5, 10, 15 and 20 have been tested. The results do not show a clear difference of performance between the cases. A depth of 15 has finally been chosen for the final setting.



	Pop = 100	Pop = 500	Pop = 1000	Pop = 2000
Max Fitness	2400	2700	3100	2800



	Gen = 50	Gen = 100	Gen = 200	Gen = 400
Max Fitness	2800	2700	3100	2800

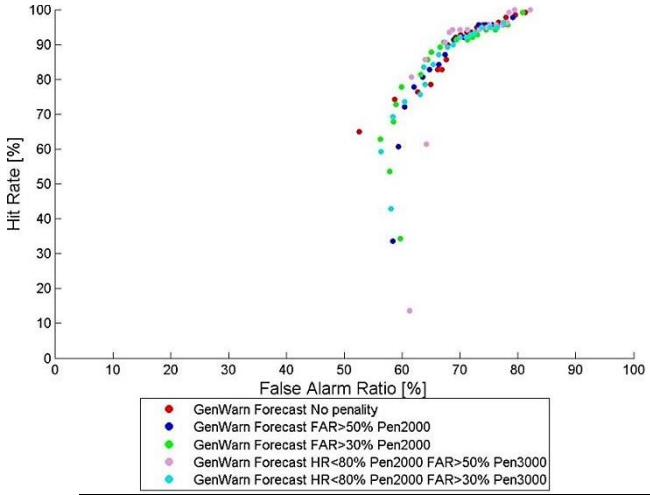


	Depth = 5	Depth = 10	Depth = 15	Depth = 20
Max Fitness	3000	2800	3100	3000

Figure 13: Verification conducted on the 2-year independent data set of the algorithms obtained for LSZG with different training parameters. On the top left, solutions with a varying population size, on the top right, solutions with a varying generations number, and at the bottom, solutions with a varying maximal depth. Each plot represents the FAR on the x-axis and HR on the y-axis as function of the probability of occurrence (0.1, 0.15, …, 0.95, 1, here not written next to each point because of the readability). Each colour corresponds to a specific setting (see the legend below the plots). The maximal fitness of each solution is reported in the table under each plot.

In order to exclude solutions with a too high FAR or a too low HR from the solutions population during the evolution, optional penalty parameters were introduced: above a given FAR (FARlim), the penalty Penfar is subtracted from the fitness value, and beneath a given HR (HRlim), the penalty Penhr is subtracted from the fitness value.

Some tests were conducted with several values for the penalty parameters. Figure 14 shows the results of the verification for the different settings. We observe that surprisingly, no significant difference results from the introduction of such penalty parameters. The maximal fitness values are nearly equal and no shifting of the pseudo ROC-curve towards lower FAR or higher HR occurs. Therefore, no penalty parameters are used in the final setting, as initially recommended by Voisard (2013).



	No penalty	FARlim=50 Penfar=2000	FARlim=30 Penfar=2000	FARlim=50% Penfar=3000 HRlim=80% Penhr=2000	FARlim=30% Penfar=3000 HRlim=80% Penhr=2000
Max Fitness	3100	3000	3100	3100	3000

Figure 14: Verification conducted on the 2-year independent data set of the algorithms obtained for LSZG with different penalty parameters. The plot represents the FAR on the x-axis and HR on the y-axis as function of the probability of occurrence (0.1, 0.15, …, 0.95, 1, here not written next to each point because of the readability). Each colour corresponds to a specific setting (see the legend below the plot). The maximal fitness of each solution is reported in the table under the plot.

Table 1: Definition of genetic parameters.

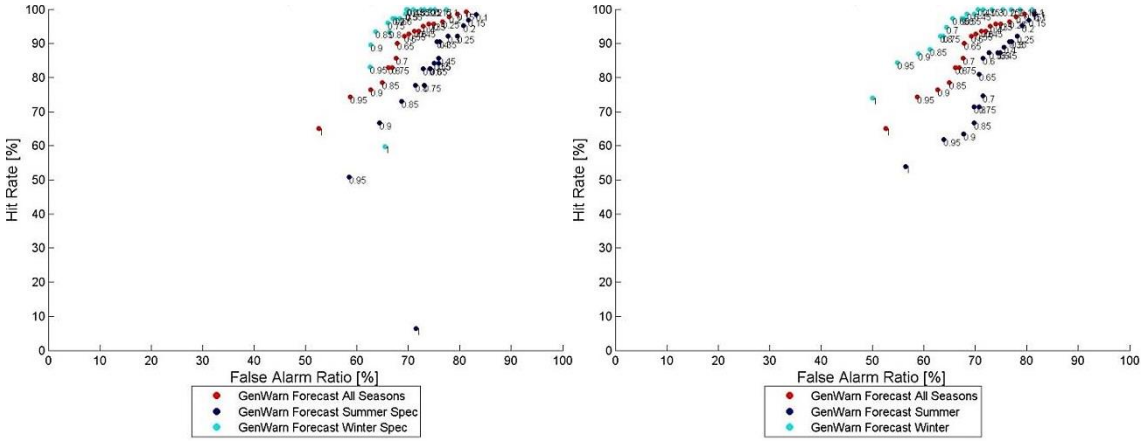
SR	Survival Rate: proportion of (best) solutions selected to build a following generation
MR	Mutation Rate: proportion of solutions that is going to be mutated
CR	Crossover Rate: proportion of solutions that is going to be crossed
Pop	Population Size: number of solutions contained in each generation
Gen	Generation Number: number of generations conducted before the evolution stops
Depth	Maximal Depth: maximal complexity of a solution
FARlim	FAR threshold above which Penfar is substracted from the fitness value
Penfar	Penality substracted from the fitness if FAR > FARlim
HRlim	HR threshold under which Penhr is substracted from the fitness value
Penhr	Penalty substracted from the fitness if HR < HRlim

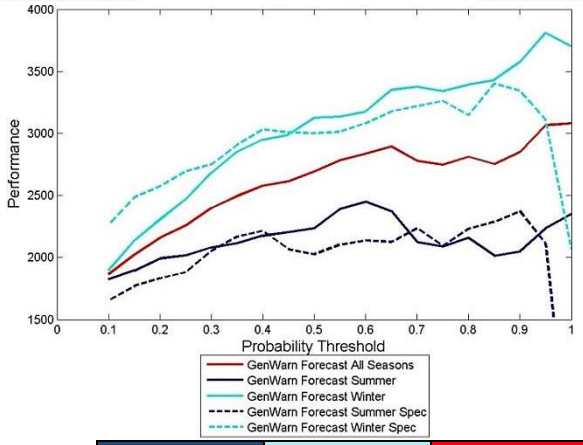
3.1.2 Winter / Summer

The meteorological phenomena leading to strong wind gusts can be quite different between summer and winter time. In summer, for example, thunderstorms represent an important origin for strong winds, which is not the case in winter. For this reason, two distinct trainings were conducted: one based on winter data (October to April) and the other one based on summer data (May to September) only. In this way, specific algorithms better adapted to each season could be obtained.

However, we can observe in Figure 15 that the season-specific algorithms do not conduct to a better performance as compared to the algorithms calculated without seasonal distinction. Both plots at the top show the system's performance with independent winter data (light blue), independent summer data (dark blue) and with the usual 2-year independent data set including both seasons (red). In the left plot, the winter and summer results represent the performance of the winter-, respectively summer-specific algorithms, whereas on the right plot, the winter and summer results represents the performance of the algorithms that were obtained without seasonal distinction. By comparing the pseudo ROC-curves, we can say that the difference in performance between the season-specific algorithms and the algorithms without seasonal distinction is not significant. This statement is confirmed when looking at the fitness evolution of both types of algorithms as a function of the probability of occurrence (plot at the bottom). The dashed lines represent the results of the season-specific algorithms, whereas the continuous lines show the results of the algorithms without seasonal distinction. We would expect the dashed lines to indicate higher fitness values than the corresponding continuous lines, but this is not the case. The fitness values of both algorithm types applied to the respective seasons are equivalent. We can conclude that the predictor data does not provide enough information for the system to be able to differentiate between summer- and winter-phenomena. Therefore, distinct seasonal trainings were no further conducted, because they did not bring any improvement in performance.

We can also observe that the GaleWarn system detects wind gusts in the winter significantly better than in the summer. The fitness difference between both seasons is high (more than 1000). Probably, not enough information concerning thunderstorms is contained in the predictors, hence the higher difficulty of the system to learn how to detect summer wind gusts correctly. This weak point should be improved in a following version by the integration of radar data. More details on this can be found in chapter 6.





	Verif. Summer		Verif. Winter		Verif. All Seasons
	Spec.	No Spec.	Spec.	No Spec.	No Spec
Max Fitness	2400	2400	3400	3800	3100

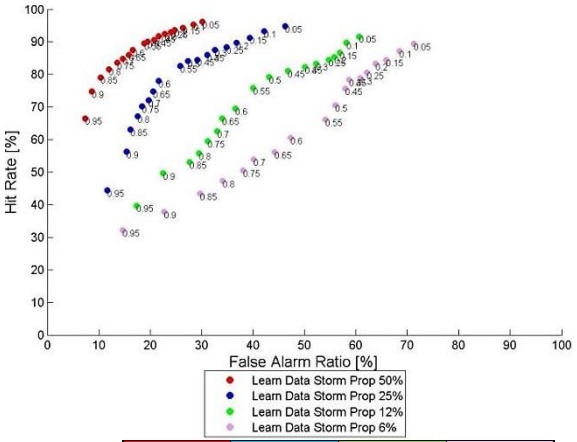
Figure 15: Both plots on the top: Verification conducted on the 2-year independent data set of the non-seasonal specific algorithms (in red) and seasonal verification conducted on the summer data only (dark blue) and on the winter data only (light blue) of the seasonal specific algorithms on the left plot and of the non-seasonal algorithms on the right plot. The plots represent the FAR on the x-axis and HR on the y-axis as function of the probability of occurrence (0.1, 0.15, …, 0.95, 1 written next to each point). Bottom plot: Fitness evolution according to the probability of occurrence. The colours have the same meaning as on the top plots. The dotted lines represent the results of the seasonal specific algorithms (like on the top left plot) and the continuous lines represent the results of the non-seasonal algorithms (like on the top right plot). The maximal fitness of each solution is reported in the table under the plots

3.1.3 Sensitivity of the Pre-Storm Proportion in the Learn-Data

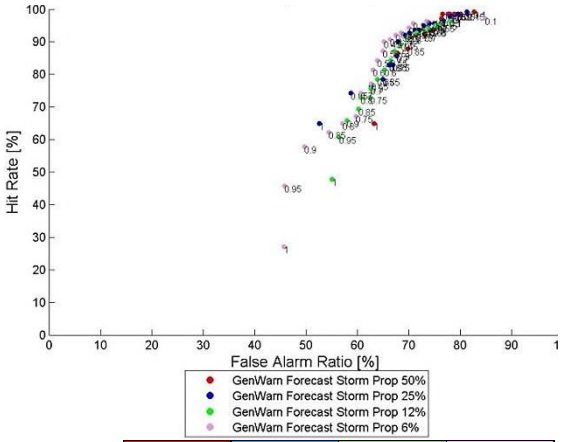
The pre-storm proportion in the learn-data has to be artificially increased in order to provide enough storm cases to the evolutionary algorithm to learn how to detect them. However, the proportion should not exceed a certain limit otherwise the tendency of giving too many alarms would be too strong. For this reason, several learning processes were conducted in which the pre-storm proportion in the learn-data set was varied. The results of these tests are presented in Figure 16, where the different colours correspond to the different pre-storm proportions in the learn-data set. The red points illustrate the performance of the algorithms that were calculated from a data set containing 50% of storm cases. The blue ones result from the algorithms that was learned with 25% of pre-

storm cases, the green ones with 12% and the purple ones with 6% of pre-storm cases (the original proportion of pre-storm cases reaches only 2.5%).

The plot on the left represents the verification results obtained by applying these algorithms to the learn-data set only. We observe that the system shows a better detection of the strong wind gusts contained in the learn-data when it learned with a high pre-storm proportion. A progressive increase in the HR and a decrease in the FAR can be seen with increasing pre-storm proportion in the learn-data set. This trend is also clear when looking at the maximal fitness values. However, if the same algorithms are applied to the 2-year independent verification data set, the results look quite different, as illustrated on the right plot. The four types of algorithms show rather poor results with fitness values varying between 2800 and 3100 only. We can conclude that the pre-storm proportion in the learn-data does not have a significant influence on the detection of strong wind gusts in independent data. Finally, a pre-storm proportion of 25% was chosen for the standard settings.



	Prop = 50%	Prop = 25%	Prop = 12%	Prop = 6%
Max Fitness	7300	6100	4600	3200



	Prop = 50%	Prop = 25%	Prop = 12%	Prop = 6%
Max Fitness	2800	3100	2900	3100

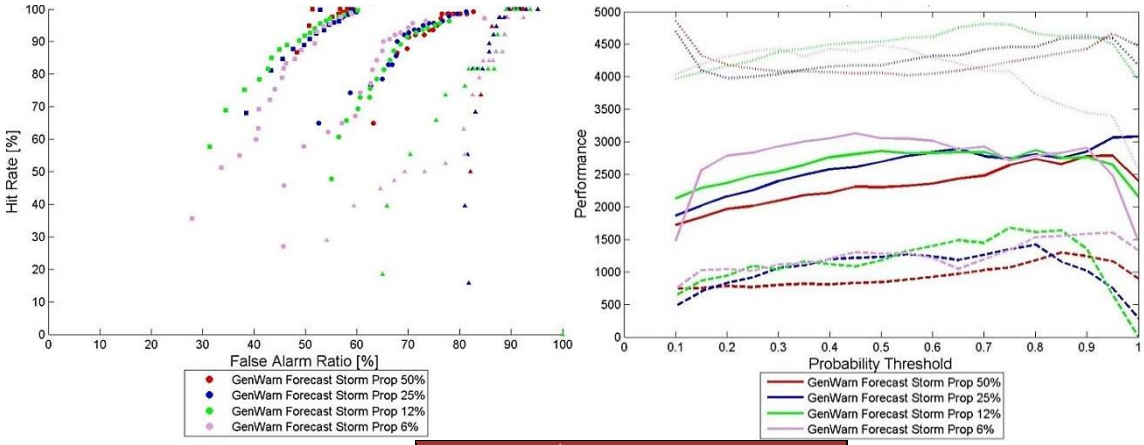
Figure 16: Left plot: Verification conducted on the learn-data of the algorithms obtained for LSZG with different pre-storm proportions in the learn-data. Right plot: Verification conducted on the 2-year independent data set of these algorithms. Both plots represent the FAR on the x-axis and HR on the y-axis as function of the probability of occurrence (0.1, 0.15, …, 0.95, 1 written next to each point). Each colour corresponds to a specific proportion of storm cases in the learn data (red 50%, blue 25%, green 12% and purple 6%). The maximal fitness of each solution is reported in the tables under the plots.

3.1.4 Sensitivity of the Wind Threshold in the Training Process

By looking at the results obtained so far, we observe that the GaleWarn system, despite training parameters and learn-data properties variation, seems to reach a limit of performance corresponding to a fitness value of 3100 (HR of about 90% for a FAR of about 65%). As a consequence, the meteorological threshold Q of 25 kt was changed to 12 kt and 33 kt in order to analyse the system performance for the detection of weaker and stronger wind gusts, respectively. Additional learning processes and verifications were therefore conducted in the same way as explained so far, but with both new meteorological thresholds 12 and 33 knots.

Figure 17 shows the verification results for the three meteorological thresholds (squares and dotted lines representing the results for Q = 12 kt, circles and continuous lines for Q = 25 kt, and triangles

and dashed lines for Q = 33 kt). We can clearly see that the performance is higher when the meteorological threshold is lower, given the same number of storm cases in the learn-data. For Q equal to 12 kt, the points are shifted to the left, towards lower FAR, and the fitness values reach 4800, which is remarkable compared to a value of only 3100 for Q equal to 25 kt. On the other hand, for Q equal to 33 kt, the points are shifted to the right, towards higher FAR, and the fitness reaches 1700 only. The major outcome of this graph is that the rarer the storm events, the more difficult it is for the system to detect them correctly. Wind gusts stronger than 25 knots already represent rare phenomena. In this case, the system shows a clear tendency of detecting too many events and hence, of producing many false alarms.



	Q = 33 kt			
	Prop = 50%	Prop = 25%	Prop = 12%	Prop = 6%
Max Fitness	1300	1400	1700	1600
	Q = 25 kt			
	Prop = 50%	Prop = 25%	Prop = 12%	Prop = 6%
Max Fitness	2800	3100	2900	3100
	Q = 12 kt			
	Prop = 50%	Prop = 25%	Prop = 12%	Prop = 6%
Max Fitness	4700	4600	4800	4500

Figure 17: Left plot: Verification conducted on the 2-year independent data set of the algorithms obtained for LSZG with different meteorological thresholds Q and different pre-storm proportions in the learn-data. The plot represents the FAR on the x-axis and HR on the y-axis as function of the probability of occurrence (0.1, 0.15, …, 0.95, 1 here not written next to each point because of the readability). Squares represent the results of the algorithms trained with Q = 12 kt, circles the results with Q = 25 kt and triangles the results with Q = 33 kt. Each colour corresponds to a specific proportion of storm cases in the learn data (red 50%, blue 25%, green 12% and purple 6%). Right plot: Fitness evolution according to the probability of occurrence. The dotted lines represent the results of the algorithms trained with Q = 12 kt, the continuous lines the results with Q = 25 kt, and the dashed lines the results with Q = 33 kt. The maximal fitness of each solution is reported in the table under the plots.

3.2 Comparison between GP and LogReg

Given the experiences presented above, the best settings for the genetic evolution (reported in Table 2) were determined and used for the calculation of every warning object. In order to get a benchmark, the logistic regression model was also calculated for each warning object based on exactly the same predictor list as for the GP method. Moreover, the performance of the direct model output (maximal wind gust) was also considered in order to assess the benefit of using a post-

processing method. The verification results of both methods and of the direct model output are illustrated in Figure 18. In the left plot, each point represents the score of a specific warning object with its optimal probability threshold. We can observe that the GP (orange) and LogReg (blue) methods are better than the direct model output (green) with HR values significantly higher. Both methods show also a similar warning performance. This is confirmed by the right plot which displays the distribution of the fitness difference between the LogReg and the GP method for each object. The red bars represent the objects for which the GP method is better, the blue bars those for which the LogReg method is better, and the white bar those for which both methods are equal. We can thus conclude that surprisingly, the GP does not bring any amelioration of the warning quality as compared to the LogReg, despite its broader solution spectrum and higher complexity.

Table 2: Final settings applied for the genetic evolution.

Survival Rate	0.4	**Maximal Depth**	15
Mutation Rate	0.4	**FARlim; Penfar**	100; 0
Crossover Rate	0.4	**HRlim; Penhr**	0; 0
# Generations	200	**Storm-% in learn-data**	25%
Population Size	1000	**Meteorological Thresh.**	25 knots

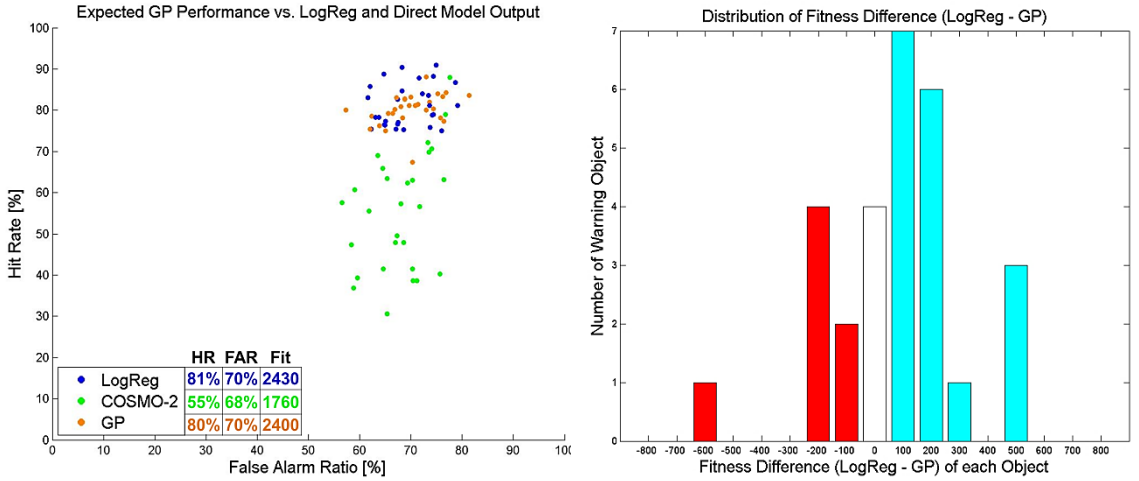


Figure 18: The left plot shows the verification results conducted on the 2-year independent data set for the GP algorithms (orange), the LogReg model (blue) and the COSMO-2 direct model output (green) for each warning object. FAR is represented on the x-axis, HR on the y-axis. Only the results of the optimal probability threshold are shown. The HR, FAR and fitness values of each method averaged on all warning objects are written in the bottom left corner. The right plot represents the distribution of the fitness difference between the LogReg and the GP method for each warning object. The red bars represent the objects for which the GP method is better, the blue bars those for which the LogReg method is better, and the white bar those for which both methods are equal.

Given that both methods deliver similar results, we decided to implement the LogReg for the operative system. First, the LogReg implies a substantially lower computing time for the model training. The GP needs one and a half days to calculate the 20 solutions of one warning object, whereas the LogReg requires only two minutes to provide the final solution (based on exactly the same predictor list). Moreover, the number of time steps contained in the learn-data is limited to 3000 in the case of GP because of the long computing time, which restricts the use of valuable information. In the case of the LogReg, the training can work with bigger data sets without reaching a

critical computing time. All available pre-storm cases can thus be considered in the training data, corresponding to 15'000 to 30'000 time steps (no-storm cases included, 25% of pre-storm cases). Finally, the use of GP would imply the implementation of a more elaborated operative system, calculating 20 complex java methods for each warning object and leading to a more expensive maintenance in case of training reiterations than the use of the LogReg.

3.3 Expected Performance of the GaleWarn System

Figure 19 shows the expected warning performance of the GaleWarn system (LogReg method[9]) for all warning objects, including those from Southern and Western Switzerland which have not been considered so far. The difference with the results of Figure 18 is that the probability threshold at which a warning is produced is now determined based on the learn-data and not on the test-data. In this way, we get a more realistic performance estimation of the system. We can observe a slight decrease in the HR compared to Figure 18 (75% instead of 81% in average), but the overall performance stays similar.

If we look at the individual objects, we notice a large performance variation between the objects. For some objects, the HR reaches almost 90% whereas for other ones it reaches only 65% for the same FAR. Two causes for this object-dependent performance could be found. First, the locations which are subject to a higher climatological storm proportion are usually better predicted (e.g. lake of Bienne BIEL, aerodrome of Sion LSGS) than the ones where strong wind gusts are less frequent (e.g. Sarnersee SARN or Hallwilersee HALL). This relation can be seen in Figure 20 where the GaleWarn fitness is plotted against the storm occurrence of each warning object. A correlation of 0.7 was calculated between these variables. This characteristic matches the statement in section 3.1.4 that the rarer the storm events, the more difficult it is for the system to detect them correctly. Secondly, the complexity of the meteorological conditions prevailing at the warning object also influences the performance of the system. The aerodrome of Samedan (LSZS) or the Haut-Lac (HAUT) are examples of locations where wind gust predictions are complicated because of the topographic situation (local wind influenced by several valleys and wind types). For these objects, despite a medium storm occurrence, the GaleWarn fitness is particularly low.

Figure 19 also contains the performance of the manual warnings issued by the forecasters (red). The scores come from the AWV1.0 tool (Weusthoff, 2012) that conducted an automatic verification of the manual warnings over the years 2011 to 2013. As explained in section 2.4, the AWV1.0 basis verification and the GaleWarn verification can be considered as comparable. The low HR appearing for the manual warnings (41% on average) confirms the fact mentioned in the introduction that storm events are frequently missed by the forecasters. The automatic system, on the other hand, shows a significantly higher HR (75% on average) than the manual warnings. GaleWarn thus reaches its goal of improving the detection of storm events. The right plot, which displays the distribution of the fitness difference between the automatic and the manual warnings for each object, corroborates this statement. The automatic warnings have a better fitness than the manual warnings for a large majority of objects. However, the high HR of GaleWarn is coupled with a high FAR (68% in average) which could represent a major problem for the forecasters.

9 Remark: similar tests like the ones described in sections 3.1.2 and 3.1.3 were also conducted with the LogReg method, leading to the same conclusions that the system is weaker in summer than in winter and that the pre-storm proportion of 25% in the learn-data is appropriate.

To sum up, the use of the GaleWarn system should lead to a general increase of the HR for every warning object. However, this improvement is strongly object-dependent: for some objects, the improvement would be huge, but for some other objects, the improvement would be only marginal. In practice, the forecaster, based on his broad experience, will have to filter the alarm proposals delivered by the system and in this way decrease the high FAR, so that the blue points in Figure 19 should be shifted to the left. The combination of man and the machine would thus improve the overall warning performance.

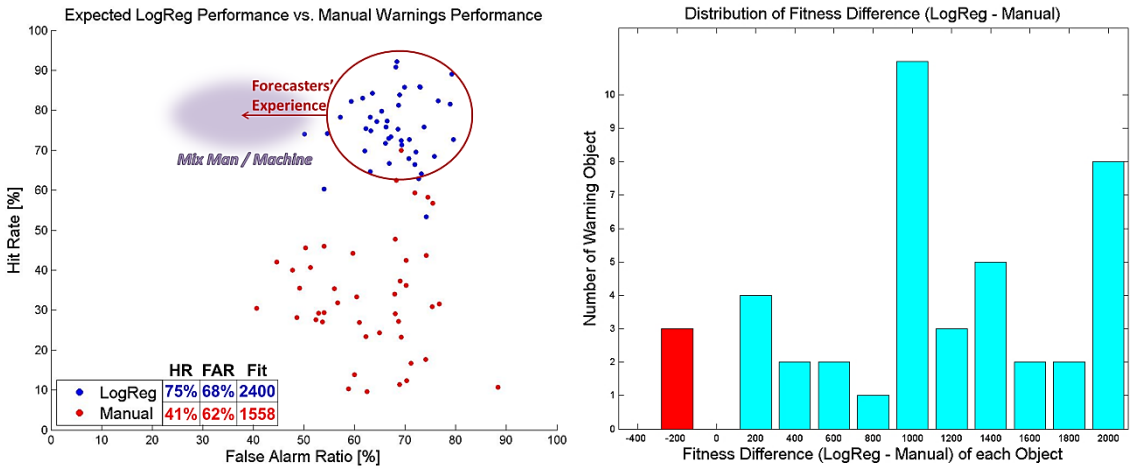


Figure 19: The left plot shows in blue the results of the verification conducted on the 2-year independent data set for the LogReg model for each warning object. Only the results of the optimal probability threshold found with the learn-data are drawn. The red points represent the results of the AWV1.0 basis verification of the manual warnings issued for each warning object over the years 2011 to 2013. FAR is represented on the x-axis, HR on the y-axis. The HR, FAR and fitness values of each method averaged on all warning objects are written in the bottom left corner. The right plot represent the distribution of the fitness difference between the LogReg and manual warnings for each object.

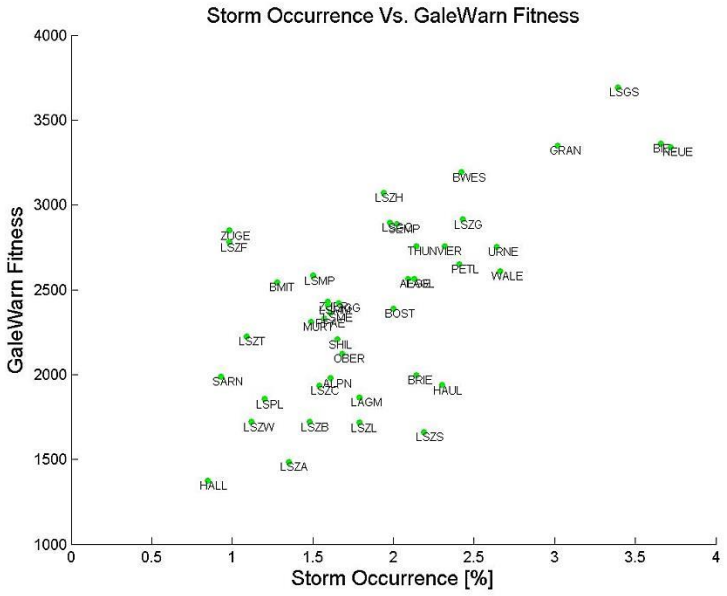


Figure 20: GaleWarn fitness as a function of the storm occurrence [%] of each warning object (abbreviation written next to each point). The storm occurrence has been calculated as the proportion of pre-storm events over all available data (4-years). The correlation between both variables is approximately 0.7.

For more details about each warning object, see Appendix B that contains a comparison between the GaleWarn system, COSMO-2 direct model output, and manual warnings for each warning object. Appendix A, which lists the warning objects name, abbreviation and number could also be of interest.

4 Real-time Tests with the Analysis Tool

The analysis tool illustrated in Figure 21 has been implemented and used for real-time tests. In this way, the weather forecaster could get accustomed to the system and its sensitivity and further analyse the warning suggestions of the GaleWarn system. The tool simply consists of a map of Switzerland, updated every 10 minutes with the latest available data, where every warning object is represented by a point. If the calculated wind gust probability of an object reaches the predefined probability threshold, the object will be displayed as a point flashing in red, indicating that a wind gust exceeding 25 knots is likely to occur within the next 30 to 180 minutes at this location. By moving the mouse over the object, the calculated probability is listed for a defined number of time steps. With a right click, the temporal evolution of the storm probability is illustrated graphically.

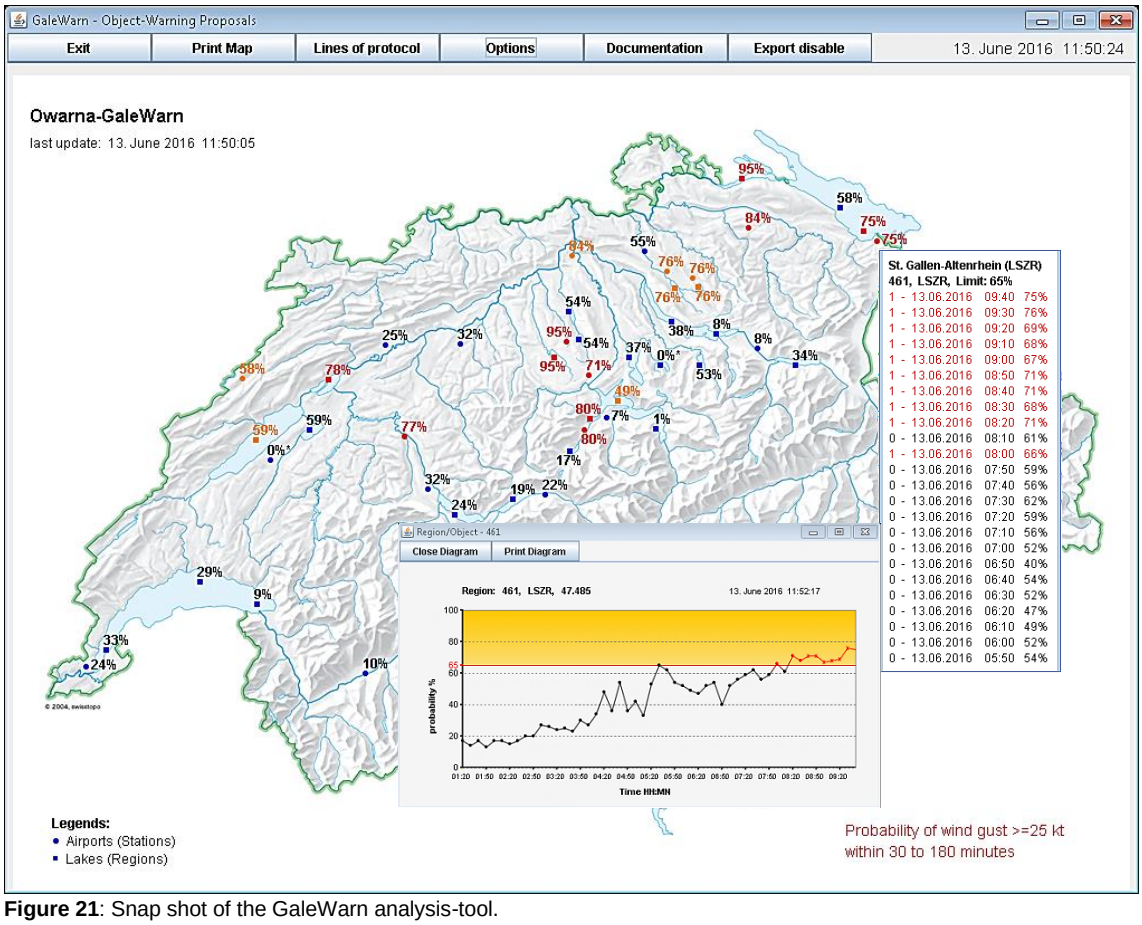


Figure 21: Snap shot of the GaleWarn analysis-tool.

4.1 Verification

Up to four months of data could be collected with the real-time tool so far. Therefore, no significant statistics can be conducted yet, but a first insight into the main tendencies and features of the GaleWarn system can be obtained.

The verification conducted here on the real-time results follows precisely the principles of the new version of the automatic warning verification method AWV2.0. The evaluation rules are more subtle than the ones presented before (AWV1.0) and differentiate not only pure hits (H), misses (M) or false alarms (FA), but also cases that are situated in between. For example, if a warning is issued when a storm event has already begun, the warning is considered as both a miss and a hit (H+M) and not only as a miss, like in AWV1.0. The following evaluation categories thus exist: H, H+FA, H+M, H+M+FA, M and FA. Their exact meaning can be found in the documentation Schraner & Reynolds (2015), section 2.3.3.

In a first step, the GaleWarn warning events must be defined. Since a GaleWarn alarm at time step t_0 means that a wind gust over 25 kt is likely to occur in the time interval $t_{0+30min}$ to t_{0+3h}, the start and the end of a warning event is not clearly set. For this reason, four types of GaleWarn warning events have been defined (illustrated in Figure 22):

1. Galetype1: warning events that already begin 30 minutes after the first alarmed time step and end 3 hours after the last alarmed time step.
2. Galetype2: warning events that also begin 30 minutes after the first alarmed time step but already end 30 minutes after the last alarmed time step.
3. Galetype3: warning events that only begin 3 hours after the first alarmed time step and also end 3 hours after the last alarmed time step.
4. Galetype4: warning events that also begin 3 hours only after the first alarmed time step but already end 30 minutes after the last alarmed time step.

All four warning types are verified in order to analyse which time scale is better adapted.

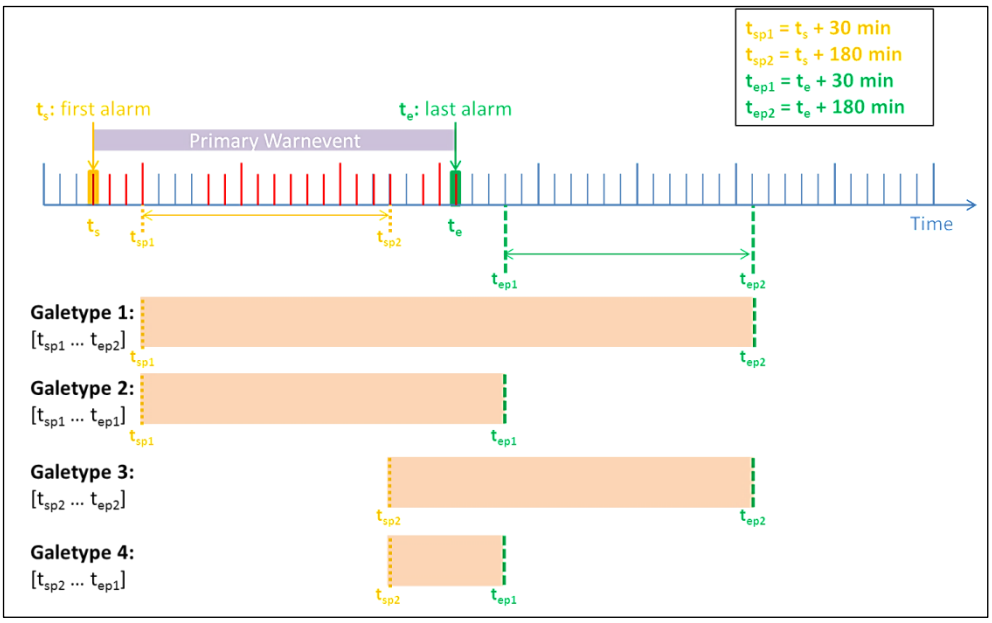


Figure 22: Illustration of the four types of GaleWarn warnings that are verified. The horizontal axis represents the time (tick every 10 minutes), and the red ticks are the time steps at which an alarm is suggested by the system.

The results of the verification conducted over the four months of available data are summarized in Table 3 and in Figure 23. The table contains the detailed scores obtained by the GaleWarn and manual warnings during this time period summed up for all warning objects. The columns in blue indicate the number of events belonging to the given evaluation categories (H, H+FA, H+M, H+M+FA, M and FA). We can observe that the warning types of GaleWarn that exhibit the best fitness values are the ones already beginning 30 minutes after the first alarm (GaleType1 and GaleType2). The HR values are significantly higher, meaning that in most alarm cases, if a strong wind gust actually arrives, it occurs within the next 3 hours after the first alarm and not afterwards. The timing of the events beginning is therefore quite well predicted. Between GaleType1 (ending 3 hours after the last alarm) and GaleType2 (ending 30 minutes after the last alarm) no significant difference exists concerning the fitness value. But the number of pure hits (H) is clearly higher for GaleType2 than for GaleType1 which, for his part, generates a higher number of hits mixed with a false alarm (H + FA), due to too long lasting warnings. Moreover, if we compare GaleType3 with GaleType4, we can note that the warning type ending already 30 minutes after the last alarm (GaleType4) has clearly a better fitness because of a lower FAR. We can, hence, deduce that the GaleWarn warning events tend to last too long. To conclude, GaleType2 represents the best manner to interpret the warning proposals: in the case of a correct warning proposal (and not a false alarm), it is important to send the warning within the 3 first hours after the first alarm, otherwise the main benefit of GaleWarn of issuing warnings sufficiently early is lost. Moreover, the warning should be ended at the latest 30 minutes after the last alarm in order to avoid too long lasting warnings.

Considering GaleType2, Table 3 shows as well that the average performance of the GaleWarn system during the real-time tests was a little better than expected with a slightly higher HR and lower FAR. As compared to the manual warnings, GaleWarn exhibits a higher averaged fitness with, as expected, a significantly higher HR (78% against 36% for the manual warnings). If we look at the absolute number of total missed events, we observe that the forecasters miss nearly four times more

events than the automatic system. On the other hand, the forecasters send almost three times less false alarms than GaleWarn.

Table 4 contains the percentage of each category of hits, misses and false alarms evaluated for the GaleWarn (GaleType2) and manual warnings of Table 3. We notice that two thirds of GaleWarn hits come from the H+FA category, which illustrates the clear tendency of GaleWarn of producing too long lasting warnings. Only 22% of the hits actually originate from pure hits that are not combined with any false alarm or missed event. This proportion is even lower for manual warnings (only 9%) for which the categories H+M and H+M+F (combination with a missed event) represent an important part of the hits, as opposed to GaleWarn. This confirms that forecasters tend to warn too late, mostly when a storm event has already begun. The manual warnings also tend to last too long (42% of the hits are from the category H+F). These too long lasting events (H+F and H+M+F) represent even 60% of the manual false alarms, meaning that the forecasters produce actually only 40% of pure false alarms. This proportion is higher for GaleWarn (62%), which proves that the automatic system tends to issue more irrational warnings than a forecasters. Finally, the proportion of missed events associated with a hit is higher for GaleWarn (37%) than for the manual warnings (27%) which shows that forecasters tend to miss entire storm events more frequently than the automatic system.

Table 3: Detailed scores obtained by the GaleWarn system and manual warnings during the period January to May 2016 over all warning objects (verification method mimic of AWV2.0). H stands for hit, FA for false alarm, M for missed event, HR for hit rate and FAR for false alarm ratio. The columns in blue indicate the number of events belonging to the specific evaluation categories (H, H+FA, H+M, H+M+FA, M and FA). The columns in green contain the total hits, misses and false alarms. The purple columns show the corresponding scores HR, FAR and fitness.

Warning Type	H	H + FA	H + M	H + M + FA	M	FA	Tot H	Tot M	Tot FA	HR	FAR	Fitness
GaleType1	44	707	23	45	121	947	819	189	1699	81%	68%	2643
GaleType2	176	541	46	39	145	964	802	230	1544	78%	66%	2657
GaleType3	26	449	29	165	313	1097	669	429	1711	61%	72%	1713
GaleType4	93	361	97	105	335	493	656	462	959	59%	60%	2384
Manual	46	219	145	107	684	214	517	936	540	36%	51%	1740

Table 4: Proportion of each category of hits, false alarms and misses evaluated for the GaleWarn (GaleType2) and manual warnings during the period January to May 2016 over all warning objects.

Warning Type	Hits				False Alarms			Misses		
	H	H + FA	H + M	H + M + FA	FA	H + F	H + M + FA	M	H + M	H + M + FA
GaleType2	22%	67%	6%	5%	62%	35%	3%	63%	20%	17%
Manual	9%	42%	28%	21%	40%	41%	20%	73%	15%	12%

In the left plot of Figure 23, HR and FAR of the GaleType2 warnings (blue) can be compared with the scores of the manual warnings (red) issued during the real-time tests for each warning object. The same patterns as in Figure 19 are easily identifiable: GaleWarn shows an overall higher HR than the manual warnings, the FAR is also high (mostly between 60% and 80%) and the warning quality strongly depends on the object. This can also be ascertained by looking at the right plot representing the distribution of the fitness difference between the automatic and the manual warnings for each object. For three objects, the manual warnings are better than the automatic ones, but for the other ones, the GaleWarn proposals show a similar or higher fitness.

We must note that the real-time tests only covered winter months, the season during which GaleWarn is known to react better. It is therefore recommended to test the real-time tool also in summer in order to get a more representative evaluation of the system. Moreover, it is difficult to assess to which extent the manual results have been influenced by the GaleWarn analysis-tool. No real warning proposals were sent so that the weather forecasters had to look actively at the tool in order to see the system's reaction. This has not been performed systematically and only by some test-users, which implies that the manual warnings have been influenced by the GaleWarn tool only marginally.

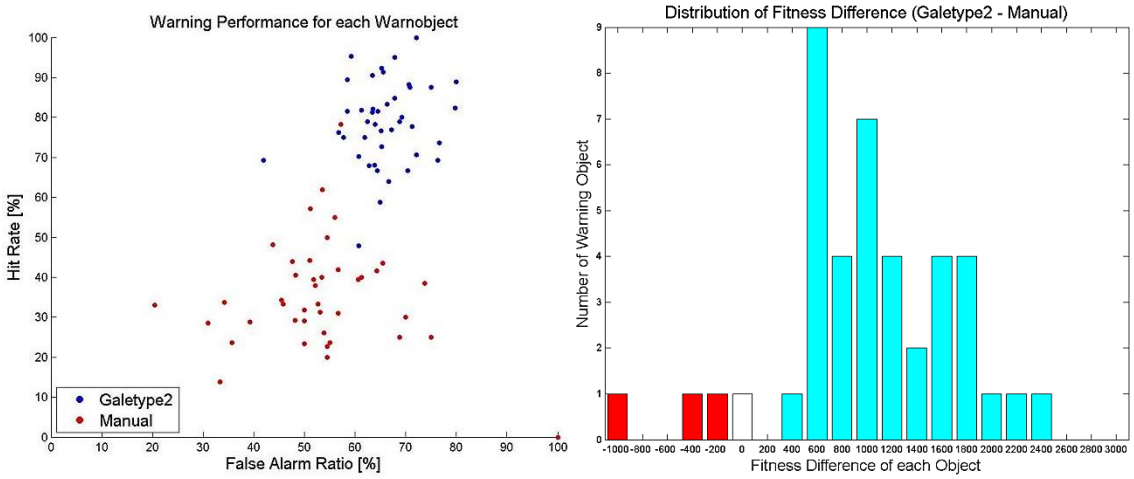


Figure 23: The left plot represents the results of the verification (mimic of AWV2.0) of the GaleWarn analysis-tool proposals (GaleType2, blue) and of the manual warnings (red) issued for each warning object over the period January to May 2016. FAR is represented on the x-axis, HR on the y-axis. The right plot shows the distribution of the fitness difference between the GaleWarn system and the manual warnings for each warning object. The red bars represent the objects for which the manual warnings are better, the blue bars those for which the GaleWarn method is better, and the white bar those for which both warning types are equal.

4.2 Forecasters' Feedbacks

The forecasters' feedbacks collected during the GaleWarn real-time tests are diverse but overall positive. The main feature that stands out is the over-sensitivity of the automatic system. However, this is often not so negatively perceived because it compensates in a way their tendency of issuing warnings too late or too sparsely.

More detailed feedbacks have given useful information about which meteorological conditions influence the general quality of the system. For example, the presence of cold air pools in the lowest atmospheric layers that prevent strong winds from reaching the ground leads to more false alarms. The COSMO-2 model does not estimate the stability parameter of the lowest layers correctly and consequently, GaleWarn mostly calculates a high wind gust probability in these cases. After this observation, a new predictor which should help the detection of such cold air lakes has been integrated in the list (temperature difference between the ground and 100 m over ground).

It also seems clear that the forecasters need to gain experience with the system before relying on its warning proposals. At the moment, they mostly prefer using well-known empirical methods than a probability calculated by a “black box” to decide if a warning should be sent or not. But for the continuous weather monitoring, GaleWarn is perceived as a great support.

5 Operative Warning Proposals

The final goal of the project is to integrate the GaleWarn system into NinJo, the workstation system used for processing and displaying meteorological data and warnings at MeteoSwiss. Thus, the forecaster will automatically receive gale warning proposals on the platform he commonly uses for weather analysis and warning edition, so that the processing of warning proposals is fully integrated in his daily tasks and can be performed properly and efficiently. Figure 24 shows an example of a gale warning proposal as it will appear automatically in NinJo (accompanied by a visual and acoustic signal). The forecaster can accept or reject the proposal. By accepting a proposal, a form containing the automatic warning information will open allowing the forecaster to edit it before sending. The start time of the warning proposal is determined by the GaleWarn algorithm whereas the end time is, by default, set to 6 hours after the start time. If the warning proposal is not processed by the forecaster after a predefined time (7 minutes), it will disappear from the list of active warning proposals and be considered as rejected (Reiniger, 2015).

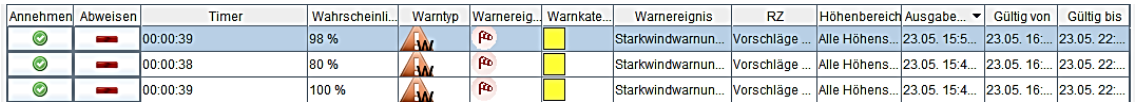


Figure 24: Example of warning proposals as they will appear automatically in NinJo. Buttons to accept or reject the proposals are on the left.

These warning suggestions can appear every 10 minutes for several objects which bears a vast potential for the forecaster to lose track of the current development. It is therefore indispensable to filter redundant proposals so that only strictly necessary ones are shown. For this reason, the following rules have been defined:

1. **Grouping of objects**
   A warning proposal can contain several objects which are grouped by the wind direction of the forecasted wind gusts. In this way, several warning objects can be warned at the time if they are subject to the same wind direction.

2. **Check versus active warnings**
   Before a warning object is included in a warning proposal, the actual warning status of the object is checked. If a warning is already active for this object and still valid for the next 30 minutes, the object will not be included in a warning proposal.

3. **Filter function**

This rule aims to decrease the number of false alarms. When the predefined probability threshold is reached for a specific object, the following conditions must be fulfilled before a warning proposal is imported into NinJo:

a. The maximal wind gust of the last hour must be larger than a critical value *ff** to send a warning proposal.

b. The system waits until the probability threshold has been exceeded *n* consecutive times before sending a warning proposal.

The first condition should be able to filter a part of the warnings that last too long (when strong wind gusts are already over), start to early (when the wind is still very calm), or are unreasonable. The second condition should help to filter the uncertain and unstable cases where the probability threshold is reached only sporadically.

The threshold *ff** can take the value 0 kt, 4 kt, 8 kt or 12 kt and *n* can be equal to 1, 2, 3 or 4. These thresholds are determined object-specifically on the learn-data according to the same method used for setting the best probability threshold described in section 2.4.

Table 5 contains the scores obtained without filter (first row) and with filter (second row), summed up over all warning objects. We can notice that the filtering does not represent an ideal way to decrease the FAR which is only reduced by approximately 1%. The HR is also slightly reduced (-3%) but stays high enough. However, if we look at the absolute difference, we observe that the application of the filter, nevertheless, prevents an important number of FA events (428) and FA time steps (11'524). We can therefore conclude that the filter function described above, despite its weak influence on the FAR, would be useful for the weather forecaster by filtering a noticeable number of FA.

Table 5: Scores obtained by the GaleWarn system without filter and with the best filter, summed up over all warning objects (verification method mimic of AWV2.0 over 2-year independent data set). The first row of red numbers corresponds to the absolute difference of scores between the results without filter and those with filter. The second row of red numbers corresponds to the relative difference.

	FA Timesteps	Tot H	Tot M	Tot FA	HR	FAR	Fitness
No Filter	113'530	3451	1180	8357	74.5 %	70.8%	2178
Best Filter	102'006	3406	1308	7929	72.3%	70.0%	2171
Abs. Diff	-11'524	-45	128	-428	-2.2%	-0.8%	-7
Rel. Diff	-10.2%	-1.3%	10.8%	-5.1%	-3.0%	-1.1%	-0.3%

4. **External control system**

The weather forecaster can selectively and temporarily turn off the production of warning proposals thanks to an external control system (prototype is shown in Figure 25). This feature is necessary if the forecaster considers for example that GaleWarn is too sensitive and does not want to receive any warning proposals for a specific lake and a certain time period. After a shift change, all objects are automatically switched on again so that the new forecaster again gets all warning proposals and can evaluate the reaction of GaleWarn himself.

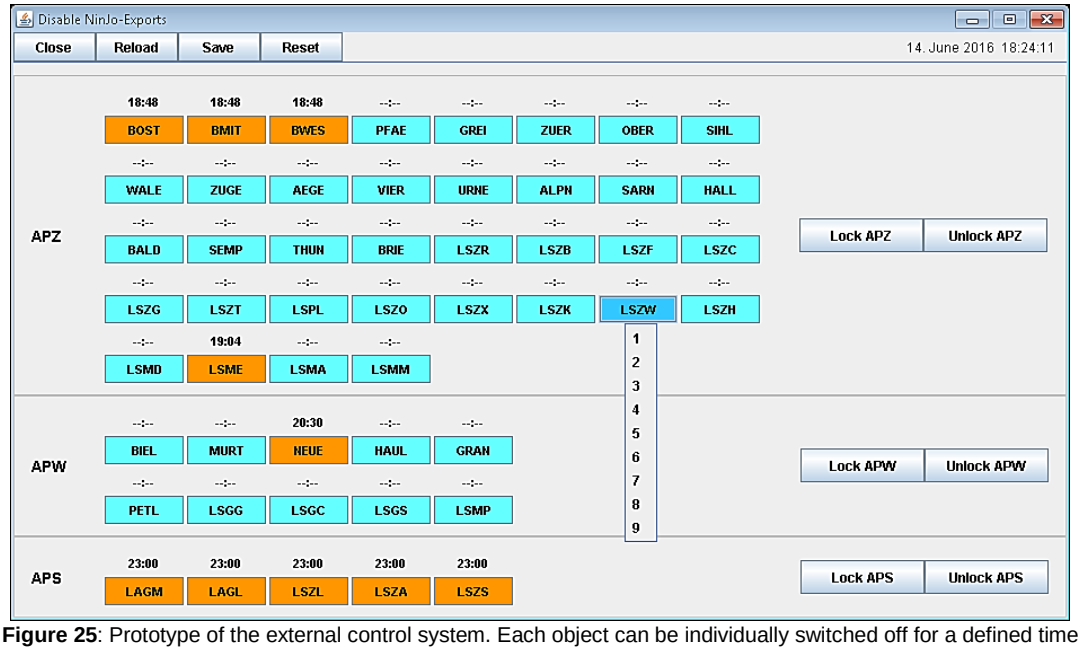


Figure 25: Prototype of the external control system. Each object can be individually switched off for a defined time (up to shift transition). The inactive objects are displayed in orange with the corresponding reset time above and the active objects are displayed in blue.

At the moment, the GaleWarn warning proposals are displayed on the NinJo test environment. The different elements listed above and their interaction (control system, check of warning status, filter function, merging of objects) must now be validated. If the tests demonstrate that the whole system runs smoothly, the operationalisation will take place in autumn 2016.

6 Conclusions and Outlook

The in-time issue of gale warnings for more than 50 lakes and aerodromes represents a big challenge for the weather forecasters. It implies a careful monitoring of the weather evolution at these numerous locations and has to be conducted beside many other forecasting tasks. Consequently, the very first strong wind gust is often missed. The aim of this OWARNA subproject was to develop a system supporting the forecasters in their ongoing weather surveillance by automatically proposing gale warnings. The system should warn the meteorologist in advance when a potential for strong wind gusts appears, so that he can send a warning in time before the first wind gust over 25 kt occurs.

In this context, the automatic warning system GaleWarn was set up, initially based on genetic programming and using relevant observation data and short-term forecasts. However, as the results obtained with the machine learning technique did not outperform the quality of logistic regression, the latter was chosen for the operationalisation because of its lower maintenance, computational cost and complexity. The GaleWarn system is expected to deliver gale warning proposals with a mean performance of about 75% HR for a 68% FAR, averaged over all warning objects. At first sight, this represents a rather poor performance because of the high FAR. But if we compare it with the actual manual warnings, we can see that a clear increase in the HR would be induced by using the GaleWarn system. The ideal solution is the combination of the human's expertise with the automatic system such that the forecasters filter the alarm proposals delivered by GaleWarn and, hence, lower the FAR, while keeping the HR higher than with the pure manual warnings alone. Combined with the forecaster's experience, the utilisation of the GaleWarn system should thus lead to a general improvement of the gale warning quality for Swiss lakes and aerodromes.

Real-time tests have been conducted during four months with the analysis-tool which consists of a Swiss map containing each warning object as a point that turns red and flashes if the storm probability reaches a predefined threshold. The results of the test-phase were satisfactory. The mean GaleWarn performance was even slightly better than expected. It has been verified that a warning should be sent within the 3 hours after the first threshold exceedance, otherwise the main benefit of GaleWarn of issuing warnings sufficiently early is lost. Forecasters delivered overall positive feedbacks despite the over-sensitivity of the system. Although they need to gain more experience with the system in order to better get to know its strengths and weaknesses, they perceive GaleWarn as a great support for the continuous weather monitoring. However, it is important to mention that the real-time tests were performed only during winter months, when the quality of GaleWarn is noticeably higher. The reaction of the system during summer should therefore be considered with attention and in case of really bad warnings, the method should be specifically improved for summer months following the propositions listed further. The same reflection should be done for specific lakes for

which the warning quality is significantly lower because of complex local climatological and topographical conditions. If the forecasters judge that the proposals are more annoying than supporting, the model should be improved for these specific lakes in order to prevent a counterproductive effect.

In operational mode, completed gale warning proposals will appear on the platform the forecasters use for the weather analysis and warning edition (NinJo), so that the processing of warning proposals is fully integrated in their daily tasks and can be performed properly and efficiently with a few clicks. In order to prevent the forecasters to get overflooded by unnecessary warning proposals, several important measures had to be taken: the merging of warning objects according to the wind direction, the inquiry of the active warning status to avoid redundant warning proposals, the implementation of a filter function and, finally, the external control system to be able to selectively and temporarily turn off the production of warning proposals. At the moment, the warning proposals are displayed on the test-environment for final validation of the different elements and their interaction. The operational production of the GaleWarn warning proposals in NinJo is planned for autumn 2016.

In a longer term, the following aspects should be considered and analysed in order to potentially improve the general performance of the GaleWarn system:

- The integration of new predictors:

  - The use of radar data in the predictor list could bring additional information about wind gusts associated with thunderstorms and hence lead to a better detection of the gusts in summer (only the nearby lightning is included in the predictor list so far). But the way of connecting the radar data to a specific object must first be defined, which already represents a complex task. The results of the work of Trefalt et al. (2015) over the analysis of convective wind gusts with radar data could be of strong interest.

  - In the same way, the integration of TRT data (Thunderstorms Radar Tracking) (Nowcasting - TRT, 2015) should also be considered. The simplest solution would be to directly use the automatic thunderstorm warning system based on the TRT algorithm which have been developed in another subproject of OWARNA Umsetzung (Kube, et al., 2016). At the moment, only thunderstorm warnings referring to warning regions are sent. But the algorithm could also produce gale warning proposal when a thunderstorm is predicted to reach a specific lake or aerodrome. This automatic system could represent a good supplement to GaleWarn during the convective season.

  - The choice of more object-specific predictors that have been carefully selected by a forecaster could help to better depict the local weather phenomena of objects subject to complex meteorological conditions.

  - Wind gusts predicted by INCA[10] do not exist yet but might be integrated in the future. In this case, it would represent a potentially excellent predictor that should be taken into account.

10 INCA (Integrated Nowcasting through Comprehensive Analysis) is a nowcasting system used at MeteoSwiss combining measurements and short-term forecasts (INCA, 2015).

  - The integration of COSMO-E[11] probability forecasts in the predictors list should be tested. It might add helpful information about the gale risk, although COSMO-E is not optimal for nowcasting purpose since only calculated twice a day.

- A better selection of the predictors should be conducted in order to remove any irrelevant predictors and, hence, facilitate the calculation of optimal solutions. At the moment, more than 50 predictors are used, which probably contributes to a less efficient model training.

- A weather-type stratified verification of the GaleWarn warnings should be conducted in order to better depict the strengths and weaknesses of the automatic system.

Finally, in September 2016, the COSMO-2 model will be replaced by the COSMO-1 model which has a resolution of 1 km instead of 2 km. Consequently, in order for GaleWarn to stay compatible with the new model, the training process for each warning object has been conducted again with the new model data. Because this task has been accomplished quite recently before the end of the report, no specific section was planned for it. The verification results based on independent past data demonstrates that the new GaleWarn version should reach a similar mean performance, averaged over all warning objects. However, the very first real-time experiences indicate a higher number of false alarms for specific lakes and aerodromes. The reason for this is still unclear but could come from the fact that the first tests have been performed in June, when GaleWarn is generally of lower quality because of the convective conditions. Another reason could be the low quality of the model data used for the new training. Indeed, the amount of training data was lower (less than 3 years) and the data came from quite different COSMO-1 model versions, which could negatively influence the training quality. The performance of the new GaleWarn version must therefore be monitored and a new training should be performed as soon as enough COSMO-1 data of the current version are available.

To conclude, the GaleWarn system obtained so far demonstrates a good potential to improve the early detection of strong wind gusts. Nevertheless, the methodology used to calculate the warning proposals could be further improved according to the points listed above. Finally, it is important to mention that this kind of post-processing tasks should be regularly refreshed, with the use of state-of-the-art data and techniques. A training reiteration should be planned regularly, in particular when new model versions are introduced.

[11] The COSMO-E model belongs to the last generation of models run by MeteoSwiss. It contains 21 forecast members with a resolution of 2.2 km and delivers probabilistic forecasts. (COSMO-E – probabilistic forecasts, 2016)

References

Ambühl, J. (2008). *Voranalyse für eine neue und sinnvollere Verifikation der Windwarnungen für die Seen.* MeteoSchweiz (intern).

Ambühl, J. (2010). *Customer oriented warning systems.* Veröffentlichungen der MeteoSchweiz.

Automatisches Messnetz. (2014, December). Retrieved September 07, 2015, from meteoswiss.admin.ch: http://www.meteoschweiz.admin.ch/home/mess-und-prognosesysteme/bodenstationen/automatisches-messnetz.html

Beobachtungsnetz für Flugwetter. (2014, December). Retrieved September 07, 2015, from meteoswiss.admin.ch: http://www.meteoschweiz.admin.ch/home/mess-und-prognosesysteme/bodenstationen/beobachtungsnetz-fuer-flugwetter.html

COSMO-E – probabilistic forecasts. (2016, March). Retrieved July 2016, from www.meteoswiss.admin.ch: http://www.meteoswiss.admin.ch/home/measurement-and-forecasting-systems/warning-and-forecasting-systems/cosmo-forecasting-system/cosmo-e-probabilistic-forecasts.html

COSMO-Prognosesystem. (2015, August). Retrieved September 07, 2015, from meteoswiss.admin.ch: http://www.meteoschweiz.admin.ch/home/mess-und-prognosesysteme/warn-und-prognosesysteme/cosmo-prognosesystem.html

Floreano, D., & Mattiussi, C. (2008). *Bio-Inspired Artificial Intelligence: Theories, Methods, and Technologies.* The MIT Press.

Hosmer, D., Lemeshow, S., & Sturdivant, R. (2013). *Applied Logistic Regression.* Wiley.

INCA. (2015, April). Retrieved June 21, 2016, from www.zamg.ac.at: http://www.zamg.ac.at/cms/de/forschung/wetter/inca

Kube, M., Ambühl, J., Appenzeller, C., Binder, P., Comment, T., Félix, C., et al. (2016). Optimierung der Warnung und Alarmierung vor Naturgefahren bei MeteoSchweiz. *Fachbericht MeteoSchweiz*, 259, 136 pp.

Meffert, K., & Rotstan, N. (2015). Retrieved June 30, 2015, from JGAP: http://jgap.sourceforge.net/

Nowcasting - TRT. (2015, December). Retrieved October 16, 2015, from meteoswiss.admin.ch: http://www.meteoswiss.admin.ch/home/measurement-and-forecasting-systems/warn-und-prognosesysteme/nowcasting.subpage.html/en/data/projects/2002/trt.html

Poli, R., Langdon, W., McPhee, N., & Koza, J. (2008). *A Field Guide to Genetic Programming.* Published via http://lulu.com and freely available at http://www.gp-field-guide.org.uk.

Reiniger, T. (2015). *Konzept: Automatische Warnvorschläge in NinJo.* MeteoSchweiz (intern).

Schraner, M., & Reynolds, S. (2015). *Automatische Warnverifikation (AWV) Version 2.0, Dokumentation der Applikation und ihrer Funktionen.* MeteoSchweiz (intern).

TAF Decoder. (2015, October). Retrieved June 15, 2016, from www.aviationweather.gov.

Trefalt, S., Martius, O., Hering, A., & Germann, U. (2015). Radar Characteristics and Patterns Related to Convective Wind Gusts in Switzerland. *Abstract for talk at Cyclone Workshop 25-30 October 2015 in Pacific Grove CA, USA.*

Voisard, R. (2010). *Genetic programming, evolutionary strategies and model output statistics to predict visibility at Zurich airport.* Bachelor Thesis University of Zurich.

Voisard, R. (2013). *GenPro:Bedienungsanleitung1_3.* MeteoSchweiz (intern).

Weusthoff, T. (2012). *Automatische Warnungsverifikation (AWV1.0): Dokumentation der Applikation und ihrer Funktionen.* MeteoSchweiz (intern).

Wilks, D. (2006). *Statistical Methods in the Atmospheric Sciences.* Elsevier.

Zimmermann, M. (2014). *Forecasting probabilistic elements of TAF based on COSMO-2 model.* Technical Report MeteoSwiss.

Züger, M. (2010). *Potential Use of Genetic Programming in Storm Warning Systems.* Bachelor Thesis University of Zurich.

Acknowledgement

We would like to thank all the people who contributed in any manner to this project.

We are especially grateful to:

- Roland Egger for the conscientious programming of the operational server part and analysis-tool.
- Roman Voisard for the programming of the evolutionary algorithm tested in this project.
- Maëlle Zimmermann for the development of the logistic regression method used in this project.
- Manuela Züger for the initial tasks carried out during her Bachelor thesis.
- Christophe Voisard for his precious support concerning all scientific and technical questions.
- Petra Baumann for her great technical support and the time spent to help us with the model data extraction.
- Thomas Reiniger for his contribution to the implementation of warning proposals in NinJo.
- Vanessa Stauch and Saskia Willemse for reviewing this report.
- Hans Romang (ex-project leader OWARNA Umsetzung) and Marco Gaia (actual project leader of OWARNA Umsetzung) for their kind support.
- All the persons who took the time to review this report or who helped us to solve technical problems.

A Appendix 1: Predictor list

Table 6: The table lists all warning objects with their full name and corresponding object number and object abbreviation. The column BAS1 contain the reference station number 1, BAS2 the reference station number 2, BAS3 the reference station number 3, HIGH the altitude station and SEC the secondary stations used for the object specific predictors calculation. If a station is added in brackets, it means that for the reference or altitude station chosen, a type of required observation data is not available. This missing data is then taken from the substitution station written in brackets. For the objects 404, 405, 470 and 474, the station DUB which is actually more representative for the objects than REH is considered for the model but not for the observation data because of unreliable measurements at DUB.

Object Number	Object Abbreviation	Object Full Name	BAS1	BAS2	BAS3	HIGH	SEC / SEC1, SEC2
401	**BOST**	**Bodensee Ost**	ARH, (STG)	-	-	HOE	TAE
402	**BMIT**	**Bodensee Mitte**	GUT	-	-	HOE	KLO
403	**BWES**	**Bodensee West**	STK, (SHA)	-	-	HOE	KLO
404	**PFAE**	**Pfäffikersee**	REH (mod DUB)	TAE	-	UEB, (SMA)	BEZ
405	**GREI**	**Greifensee**	REH (mod DUB)	TAE	-	UEB, (SMA)	BEZ
406	**ZUER**	**Zürichsee**	WAE	-	-	UEB, (SMA)	KLO
407	**OBER**	**Obersee**	SCM, (WAE)	-	-	HOE	SMA
408	**SHIL**	**Sihlsee**	EIN	-	-	UEB, (SMA)	SMA
409	**WALE**	**Walensee**	QUI, (GLA)	-	-	HOE	SMA, VAD
410	**ZUGE**	**Zugersee**	CHZ, (LUZ)	-	-	NAP, (PIL)	BUS
411	**AEGE**	**Ägerisee**	AEG, (EIN)	-	-	NAP, (PIL)	BUS, ALT
412	**VIER**	**Vierwaldstättersee**	LUZ	BUO	-	NAP, (PIL)	BUS
413	**URNE**	**Urnersee**	ALT	-	-	GUE	LUZ
414	**ALPN**	**Alpnachersee**	LUZ	-	-	NAP, (PIL)	EGO
415	**SARN**	**Sarnersee**	GIH, (LUZ)	-	-	NAP, (PIL)	WYN
416	**HALL**	**Hallwilersee**	MOA, (LUZ)	-	-	NAP, (PIL)	GRE
417	**BALD**	**Baldeggersee**	MOA, (LUZ)	-	-	NAP, (PIL)	GRE
418	**SEMP**	**Sempachersee**	EGO, (LUZ)	-	-	NAP, (PIL)	GRE
419	**THUN**	**Thunersee**	THU, (BER)	INT	-	BAN, (GRH)	BER
420	**BRIE**	**Brienzersee**	INT, (BRZ,MER)	BRZ	-	GRH	BER
461	**LSZR**	**St. Gallen - Altenrhein**	ARH, (STG)	-	-	HOE	TAE
462	**LSZB**	**Bern - Belp**	BRN, (BER)	-	-	BAN, (PLF)	GRE
463	**LSZF**	**Birrfeld**	BUS	-	-	RUE	BAS
464	**LSZC**	**Buochs**	BUO, (ALT)	-	-	ENG	LUZ
465	**LSZG**	**Grenchen**	GRE, (WYN)	-	-	CHA	MAH
466	**LSZT**	**Lommis**	TAE	-	-	HOE	BUS
467	**LSPL**	**Langenthal**	WYN	-	-	NAP	BER
468	**LSZO**	**Luzern - Beromünster**	EGO, (LUZ)	-	-	NAP, (PIL)	GRE
469	**LSZX**	**Schänis**	SCM, (WAE)	-	-	HOE	SMA
470	**LSZK**	**Speck - Fehraltorf**	REH (mod DUB)	TAE	-	UEB, (SMA)	BEZ
471	**LSZW**	**Thun**	THU, (BER)	-	-	PLF	BER
472	**LSZH**	**Zürich - Kloten**	KLO	REH	-	LAE	WYN
474	**LSMD**	**Dübendorf**	REH (mod DUB)	TAE	-	UEB, (SMA)	BEZ
475	**LSME**	**Emmen**	LUZ	-	-	NAP, (PIL)	BUS
476	**LSMA**	**Alpnach**	LUZ	-	-	NAP, (PIL)	EGO
477	**LSMM**	**Meiringen**	MER	-	-	GRH	BER

Object Number	Object Abbreviation	Object Full Name	BAS1	BAS2	BAS3	HIGH	SEC / SEC1, SEC2
431	**BIEL**	**Lac de Bienne**	NEU	CRM	-	CHA	KLO, OTL, GRE
432	**MURT**	**Lac de Morat**	PAY	-	-	CHA	KLO, OTL
433	**NEUE**	**Lac de Neuchâtel**	NEU	MAR	-	CHA	KLO, OTL
434	**HAUL**	**Haut Lac**	BOU (AIG)	-	-	DOL	PUY, AIG, ATT
435	**GRAN**	**Grand Lac**	PUY	PRE	CGI	DOL	KLO, BIE, GVE
436	**PETL**	**Petit Lac**	GVE	CGI	-	DOL	KLO, OTL, BIE
480	**LSGG**	**Genève – Cointrin**	GVE	-	-	DOL	KLO, BIE
481	**LSGC**	**Les Esplatures**	CDF	-	-	CHA	KLO, OTL
482	**LSGS**	**Sion**	SIO	-	-	MVE	PUY, OTL, CHA
483	**LSMP**	**Payerne**	PAY	-	-	CHA	KLO MAH, OTL, MUB
451	**LAGM**	**Lago Maggiore**	MAG	OTL	-	CIM	COM, LUG
452	**LAGL**	**Lago di Lugano**	LUG	MAG	-	GEN	SBO, COM
490	**LSZL**	**Locarno**	MAG	OTL	-	CIM	COM, LUG
491	**LSZA**	**Lugano – Agno**	AGN (LUG)	-	-	GEN	COM, SBO
492	**LSZS**	**Samedan**	SAM	-	-	COV	CIM, NAP

Table 7: List of the observation data extracted for the standard predictors calculation. The station abbreviations BAS1, BAS2, BAS3 and HIGH refer to the object specific stations contained in Table 6. RHW corresponds to the station Rheinwaldhorn, for which the weather type is calculated.

Short Name	ID	Parameter	Symbol	Unit	Station
prestas0	90	Atmospheric pressure at station height (QFE); momentary value	**qf**	hPa	**BAS1**
tre200s0	91	Air temperature 2 m over ground; momentary value	**tt**	°C	**BAS1**
tre005s0	92	Air temperature 5 cm over grass; momentary value	**t5**	°C	**BAS1**
tpo200s0	733	Potential temperature 2 m over ground; momentary value	**pt**	°C	**BAS1**
tre205sd	1032	Difference air temperature 5 cm over grass – 2m over ground; momentary value	**td**	°C	**BAS1**
sre000z0	94	Sunshine duration; 10-minutes sum	**ss**	min	**BAS1**
brecloz0	99	Nearby lightning (less than 3 km)	**de**	-	**BAS1**
tde200s0	194	Dew point 2m over ground; momentary value	**tp**	°C	**BAS1**
rre150z0	93	Precipitation ; 10-minutes sum	**rr**	mm	**BAS1**
dkl010z0	197	Wind direction ; 10 minutes average	**dd**	°	**BAS1**
fu2010z0	1077	Wind speed; 10-minutes average	**fm**	kt	**BAS1**
fu2010z1	1934	Wind gust (1 second); maximum	**ffb1**	kt	**BAS1**
fu2010z1	1934	Wind gust (1 second); maximum	**ffb2**	kt	**BAS2**
fu2010z1	1934	Wind gust (1 second); maximum	**ffb3**	kt	**BAS3**
tre200s0	91	Air temperature 2 m over ground; momentary value	**tth**	°C	**HIGH**
tpo200s0	733	Potential temperature 2 m over ground; momentary value	**pth**	°C	**HIGH**
dkl010z0	197	Wind direction ; 10 minutes average	**ddh**	°	**HIGH**
fu2010z1	1934	Wind gust (1 second); maximum	**ffh**	kt	**HIGH**
fu2010z0	1077	Wind speed; 10-minutes average	**fmh**	kt	**HIGH**
wkmowkd0	4795	GWTWS-Weather Types	**wtype**	-	**RHW**

Table 8: The table contains the calculated standard predictors derived from the observation data listed in Table 7.

Original Data	Derived Predictor	Description
qf	qfe	QFE at time t_0 $= qf(t_0)$
	qfdif	Maximum of (difference qfe - af at time $t_{0\text{-}10min}$, $t_{0\text{-}20min}$, $t_{0\text{-}30min}$, $t_{0\text{-}40min}$, $t_{0\text{-}50min}$, $t_{0\text{-}60min}$)^2 $= max((qf(t_0)-qf(t_{0\text{-}10min}))^2, (qf(t_0)-qf(t_{0\text{-}20min}))^2, (qf(t_0)-qf(t_{0\text{-}30min}))^2, (qf(t_0)-qf(t_{0\text{-}40min}))^2, (qf(t_0)-qf(t_{0\text{-}50min}))^2, (qf(t_0)-qf(t_{0\text{-}60min}))^2)$
tt	ttt	2m temperature at time t_0 $= tt(t_0)$
	ttdif	Maximum of (absolute difference ttt - tt at time $t_{0\text{-}10min}$, $t_{0\text{-}20min}$, $t_{0\text{-}30min}$, $t_{0\text{-}40min}$, $t_{0\text{-}50min}$, $t_{0\text{-}60min}$) $= max(abs(tt(t_0)-tt(t_{0\text{-}10min})), abs(tt(t_0)-tt(t_{0\text{-}20min})), abs(tt(t_0)-tt(t_{0\text{-}30min})), abs(tt(t_0)-tt(t_{0\text{-}40min})), abs(tt(t_0)-tt(t_{0\text{-}50min})), abs(tt(t_0)-tt(t_{0\text{-}60min})))$
t5	t5c	Temperature 5 cm over grass at time t_0 $= t5(t_0)$
	t5cdif	Maximum of (absolute difference t5c - t5 at time $t_{0\text{-}10min}$, $t_{0\text{-}20min}$, $t_{0\text{-}30min}$, $t_{0\text{-}40min}$, $t_{0\text{-}50min}$, $t_{0\text{-}60min}$) $= max(abs(t5(t_0)-t5(t_{0\text{-}10min})), abs(t5(t_0)-t5(t_{0\text{-}20min})), abs(t5(t_0)-t5(t_{0\text{-}30min})), abs(t5(t_0)-t5(t_{0\text{-}40min})), abs(t5(t_0)-t5(t_{0\text{-}50min})), abs(t5(t_0)-t5(t_{0\text{-}60min})))$
td	tdif	Difference temperature 5 cm over grass – 2m over ground at time t_0 $= td(t_0)$
	tddif	Maximum of (absolute difference tdif - td at time $t_{0\text{-}10min}$, $t_{0\text{-}20min}$, $t_{0\text{-}30min}$, $t_{0\text{-}40min}$, $t_{0\text{-}50min}$, $t_{0\text{-}60min}$) $= max(abs(td(t_0)-td(t_{0\text{-}10min})), abs(td(t_0)-td(t_{0\text{-}20min})), abs(td(t_0)-td(t_{0\text{-}30min})), abs(td(t_0)-td(t_{0\text{-}40min})), abs(td(t_0)-td(t_{0\text{-}50min})), abs(td(t_0)-td(t_{0\text{-}60min})))$
de	deac	Sum of near lightning over the last 20 minutes at time t_0 $= de(t_0)+de(t_{0\text{-}10min})$
ss	sss	Sunshine of the last 10 minutes at time t_0 $= ss(t_0)$
	ss60ac	Sunshine of the last 70 minutes at time t_0 $= ss(t_0) + ss(t_{0\text{-}10min}) + ss(t_{0\text{-}20min}) + ss(t_{0\text{-}30min}) + ss(t_{0\text{-}40min}) + ss(t_{0\text{-}50min}) + ss(t_{0\text{-}60min})$
tp	tpt	Dew point at time t_0 $= tp(t_0)$
	tpdif	Maximum of (absolute difference tpt - tp at time $t_{0\text{-}10min}$, $t_{0\text{-}20min}$, $t_{0\text{-}30min}$, $t_{0\text{-}40min}$, $t_{0\text{-}50min}$, $t_{0\text{-}60min}$) $= max(abs(tp(t_0)-tp(t_{0\text{-}10min})), abs(tp(t_0)-tp(t_{0\text{-}20min})), abs(tp(t_0)-tp(t_{0\text{-}30min})), abs(tp(t_0)-tp(t_{0\text{-}40min})), abs(tp(t_0)-tp(t_{0\text{-}50min})), abs(tp(t_0)-tp(t_{0\text{-}60min})))$
rr	rrr	Precipitation sum of the last hour at time t0 $= rr(t_0) + rr(t_{0\text{-}10min}) + rr(t_{0\text{-}20min}) + rr(t_{0\text{-}30min}) + rr(t_{0\text{-}40min}) + rr(t_{0\text{-}50min}) + rr(t_{0\text{-}60min})$
dd	ddd	Average of the mean wind direction at time t_0, $t_{0\text{-}10min}$, $t_{0\text{-}20min}$, $t_{0\text{-}30min}$, $t_{0\text{-}40min}$, $t_{0\text{-}50min}$, $t_{0\text{-}60min}$ $= (tan^{-1}(\frac{\$mean\_sin}{\$mean\_cos}) * \frac{360}{2*\pi}) \; mod \; 360$ With $\$mean\_sin = (sin(dd_0*2\pi/360)+ sin(dd_1*2\pi/360)+ sin(dd_2*2\pi/360)+ sin(dd_3*2\pi/360)+ sin(dd_4*2\pi/360)+ sin(dd_5*2\pi/360)+ sin(dd_6*2\pi/360))/7$ $\$mean\_cos = (cos(dd_0*2\pi/360)+ cos(dd_1*2\pi/360)+ cos(dd_2*2\pi/360)+ cos(dd_3*2\pi/360)+ cos(dd_4*2\pi/360)+ cos(dd_5*2\pi/360)+ cos(dd_6*2\pi/360))/7$ $dd_0 = dd(t_0)$ $dd_1 = dd(t_{0\text{-}10min})$ $dd_2 = dd(t_{0\text{-}20min})$ $dd_3 = dd(t_{0\text{-}30min})$ $dd_4 = dd(t_{0\text{-}40min})$ $dd_5 = dd(t_{0\text{-}50min})$ $dd_6 = dd(t_{0\text{-}60min})$
fm	fm0	Wind speed (10-min average) at time t_0 $= fm(t_0)$
	fmm	Average wind speed of the last hour at time t_0 (from 10-min average) $= (fm0+ fm1+ fm2+ fm3+ fm4+ fm5+fm6)/7$ With $fm1 = fm(t_{0\text{-}10min})$ $fm2 = fm(t_{0\text{-}20min})$ $fm3 = fm(t_{0\text{-}30min})$ $fm4 = fm(t_{0\text{-}40min})$ $fm5 = fm(t_{0\text{-}50min})$ $fm6 = fm(t_{0\text{-}60min})$
ffb1, ffb2, ffb3	ff0	Maximal wind gust (1 sec integration) of the last 10 minutes at time t_0 of the stations BAS1, BAS2 and BAS3 $= max(ffb1(t_0), ffb2(t_0), ffb3(t_0))$

	ffma	Maximal wind gust (1 sec integration) of the last hour at time t_0 of the stations BAS1, BAS2 and BAS3 ***= max(ff0,ff1,ff2,ff3,ff4,ff5,ff6)*** *With* $ff1 = max(ffb1(t_{0\text{-}10min}), ffb2(t_{0\text{-}10min}), ffb3(t_{0\text{-}10min}))$ $ff2 = max(ffb1(t_{0\text{-}20min}), ffb2(t_{0\text{-}20min}), ffb3(t_{0\text{-}20min}))$ $ff3 = max(ffb1(t_{0\text{-}30min}), ffb2(t_{0\text{-}30min}), ffb3(t_{0\text{-}30min}))$ $ff4 = max(ffb1(t_{0\text{-}40min}), ffb2(t_{0\text{-}40min}), ffb3(t_{0\text{-}40min}))$ $ff5 = max(ffb1(t_{0\text{-}50min}), ffb2(t_{0\text{-}50min}), ffb3(t_{0\text{-}50min}))$ $ff6 = max(ffb1(t_{0\text{-}60min}), ffb2(t_{0\text{-}60min}), ffb3(t_{0\text{-}60min}))$
pth, pt	**ptdif**	Difference of 2m dew point at time t_0 between the altitude station HIGH and the reference station BAS1 ***= pth(t_0)-pt(t_0)***
tth	**ttdh**	2m temperature difference between time t_0 and time $t_{0\text{-}30min}$ at station HIGH ***= tth(t_0)-tth($t_{0\text{-}30min}$)***
ddh	**dddh**	Average of the mean wind direction at time t_0, $t_{0\text{-}10min}$, $t_{0\text{-}20min}$, $t_{0\text{-}30min}$, $t_{0\text{-}40min}$, $t_{0\text{-}50min}$, $t_{0\text{-}60min}$ at station HIGH $= (tan^{-1}(\frac{\$mean\_sin}{\$mean\_cos}) * \frac{360}{2 * \pi}) \; mod \; 360$ *With* $\$mean\_sin = (sin(ddh_0*2\pi/360)+ sin(ddh_1*2\pi/360)+ sin(ddh_2*2\pi/360)+ sin(ddh_3*2\pi/360)+ sin(ddh_4*2\pi/360)+ sin(ddh_5*2\pi/360)+ sin(ddh_6*2\pi/360))/7$ $\$mean\_cos = (cos(ddh_0*2\pi/360)+ cos(ddh_1*2\pi/360)+ cos(ddh_2*2\pi/360)+ cos(ddh_3*2\pi/360)+ cos(ddh_4*2\pi/360)+ cos(ddh_5*2\pi/360)+ cos(ddh_6*2\pi/360))/7$ $ddh_0 = ddh(t_0)$ $ddh_1 = ddh(t_{0\text{-}10min})$ $ddh_2 = ddh(t_{0\text{-}20min})$ $ddh_3 = ddh(t_{0\text{-}30min})$ $ddh_4 = ddh(t_{0\text{-}40min})$ $ddh_5 = ddh(t_{0\text{-}50min})$ $ddh_6 = ddh(t_{0\text{-}60min})$
fmh	**fm0h**	Wind speed (10-min average) at time t_0 at station HIGH ***= fmh(t_0)***
	fmmh	Average wind speed of the last hour at time t_0 (from 10-min average) at station HIGH ***= (fm0h+ fm1h+ fm2h+ fm3h+ fm4h+ fm5h+fm6h)/7*** *With* $fm1h = fmh(t_{0\text{-}10min})$ $fm2h = fmh(t_{0\text{-}20min})$ $fm3h = fmh(t_{0\text{-}30min})$ $fm4h = fmh(t_{0\text{-}40min})$ $fm5h = fmh(t_{0\text{-}50min})$ $fm6h = fmh(t_{0\text{-}60min})$
ffh	**ff0h**	Maximal wind gust (1 sec integration) of the last 10 minutes at time t_0 at station HIGH ***= ffh(t_0)***
	ffmah	Maximal wind gust (1 sec integration) of the last hour at time t_0 at station HIGH ***= max(ff0h,ff1h,ff2h,ff3h,ff4h,ff5h,ff6h)*** *With* $ff1h = ffh(t_{0\text{-}10min})$ $ff2h = ffh(t_{0\text{-}20min})$ $ff3h = ffh(t_{0\text{-}30min})$ $ff4h = ffh(t_{0\text{-}40min})$ $ff5h = ffh(t_{0\text{-}50min})$ $ff6h = ffh(t_{0\text{-}60min})$
wtype	**wtype**	Weather Type at time t_0 ***= wtype(t_0)***
stype	**stype**	Season Type at time t_0 (binary: summer = 1 or winter = 0) ***= 1 if 5 ≤ month(t_0) ≤ 9 (Summer from Mai to September)*** ***= 0 if month(t_0) < 5 or month(t_0) > 9 (Winter from October to April)***

Table 9: List of the COSMO-2 forecasts used for the standard predictors calculation. The station abbreviations BAS1 and HIGH refer to the object specific stations contained in Table 6.

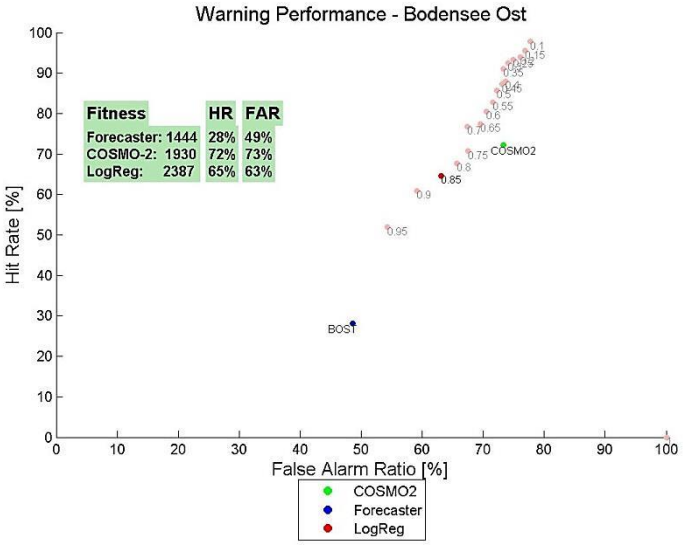
Short Name	ID	Parameter	Symbol	Unit	Station
DD_10M	2257	Hourly average wind direction at 10m	**DD**	°	**BAS1**
FF_10M	2261	Hourly average wind speed at 10m	**FF**	m/s	**BAS1**
VMAX_10M	2262	Maximum wind gusts at 10m	**VV**	m/s	**BAS1**
WSHEAR_0-3km	6042	Wind shear between surface and 3 km	**WS**	m/s	**BAS1**
CAPE_MU	6043	Convective available potential energy of most unstable parcel	**CA**	J/kg	**BAS1**
T_500	6203	Temperature at 500 hPa	**T500**	°C	**BAS1**
T_850	6202	Temperature at 850 hPa	**T850**	°C	**BAS1**
DD_500	6214	Hourly average wind direction at 500 m above ground	**DD5**	°	**BAS1**
FF_500	6215	Hourly average wind speed at 500 m above ground	**FF5**	m/s	**BAS1**
DD_1000	6216	Hourly average wind direction at 1000 m above ground	**DD10**	°	**BAS1**
FF_1000	6217	Hourly average wind speed at 1000 m above ground	**FF10**	m/s	**BAS1**
BRN_200_2000	6218	Bulk Richardson Number: Integral between 200 m and 2000 m over ground	**BRN**	-	**BAS1**
THETA_2M	5398	2m potential temperature	**PT**	°C	**BAS1**
T2M	2285	2m temperature	**TT**	°C	**BAS1**
T_100M	7391	Temperature at 100 m above ground	**T100**	°C	**BAS1**
DBZ_CMAX	7392	Unattenuated radar reflectivity in Rayleigh approximation: column maximum	**DBZ**	-	**BAS1**
VMAX_10M	2262	Maximum wind gusts at 10m	**VVH**	m/s	**HIGH**
THETA_2M	5396	2m potential temperature	**PTH**	K	**HIGH**

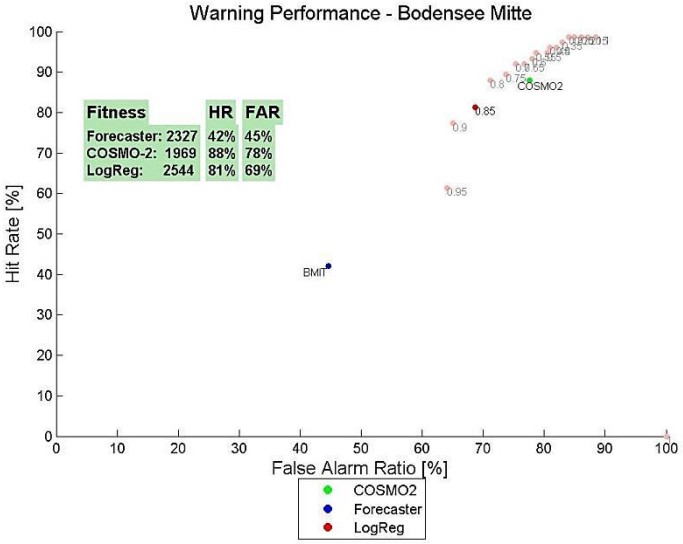
Table 10: The table contains the calculated standard predictors derived from the COSMO-2 forecasts listed in Table 9.

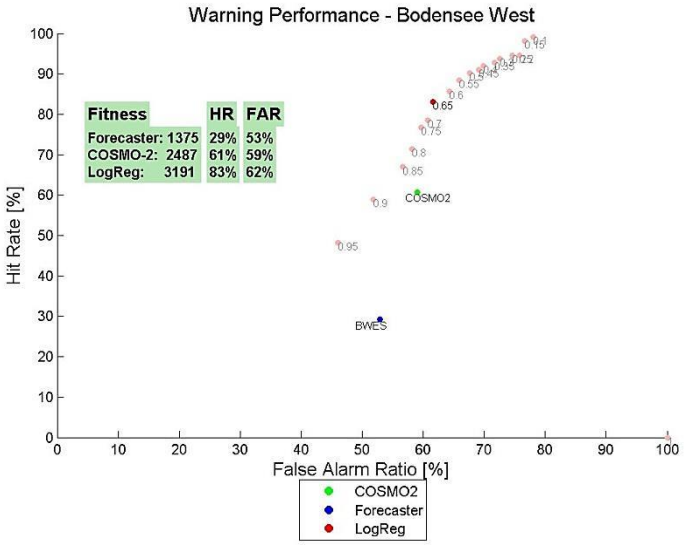
Original Data	Derived Predictor	Description
DD	**dmo**	Average of DD_10M forecasts at time t_{0+1h}, t_{0+2h}, t_{0+3h} $= (tan^{-1}(\frac{\$mean\_sin}{\$mean\_cos}) * \frac{360}{2*\pi}) \; mod \; 360$ *With* $\$mean\_sin = (sin(DD_1*2\pi/360)+sin(DD_2*2\pi/360)+sin(DD_3*2\pi/360))/3$ $\$mean\_cos = (cos(DD_1*2\pi/360)+cos(DD_2*2\pi/360)+cos(DD_3*2\pi/360))/3$ $DD_1 = DD(t_{0+1h})$ $DD_2 = DD(t_{0+2h})$ $DD_3 = DD(t_{0+3h})$
DD5	**dmo5**	Average of DD5 forecasts at time t_{0+1h}, t_{0+2h}, t_{0+3h} $= (tan^{-1}(\frac{\$mean\_sin}{\$mean\_cos}) * \frac{360}{2*\pi}) \; mod \; 360$ *With* $\$mean\_sin = (sin(DD_1*2\pi/360)+sin(DD_2*2\pi/360)+sin(DD_3*2\pi/360))/3$ $\$mean\_cos = (cos(DD_1*2\pi/360)+cos(DD_2*2\pi/360)+cos(DD_3*2\pi/360))/3$ $DD_1 = DD5(t_{0+1h})$ $DD_2 = DD5(t_{0+2h})$ $DD_3 = DD5(t_{0+3h})$
DD10	**dmo10**	Average of DD10 forecasts at time t_{0+1h}, t_{0+2h}, t_{0+3h} $= (tan^{-1}(\frac{\$mean\_sin}{\$mean\_cos}) * \frac{360}{2*\pi}) \; mod \; 360$ *With* $\$mean\_sin = (sin(DD_1*2\pi/360)+sin(DD_2*2\pi/360)+sin(DD_3*2\pi/360))/3$ $\$mean\_cos = (cos(DD_1*2\pi/360)+cos(DD_2*2\pi/360)+cos(DD_3*2\pi/360))/3$ $DD_1 = DD10(t_{0+1h})$ $DD_2 = DD10(t_{0+2h})$ $DD_3 = DD10(t_{0+3h})$
FF	**fmo**	Maximum of FF_10M forecasts *** 1.94** (in kt) at time t_{0+1h}, t_{0+2h}, t_{0+3h} $= max(FF(t_{0+1h}), FF(t_{0+2h}), FF(t_{0+3h}))*1.94$
FF5	**fmo5**	Maximum of FF5 forecasts *** 1.94** (in kt) at time t_{0+1h}, t_{0+2h}, t_{0+3h} $= max(FF5(t_{0+1h}), FF5(t_{0+2h}), FF5(t_{0+3h}))*1.94$
FF10	**fmo10**	Maximum of FF10 forecasts *** 1.94** (in kt) at time t_{0+1h}, t_{0+2h}, t_{0+3h} $= max(FF10(t_{0+1h}), FF10(t_{0+2h}), FF10(t_{0+3h}))*1.94$
VV	**mmo**	Maximum of VMAX_10M forecasts *** 1.94** (in kt) at time t_{0+1h}, t_{0+2h}, t_{0+3h} $= max(VV(t_{0+1h}), VV(t_{0+2h}), VV(t_{0+3h}))*1.94$
WS	**wshe**	Maximum of WSHEAR_0-3km forecasts at time t_{0+1h}, t_{0+2h}, t_{0+3h} $= max(WS(t_{0+1h}), WS(t_{0+2h}), WS(t_{0+3h}))$
CA	**cape**	Maximum of CAPE_MU forecasts at time t_{0+1h}, t_{0+2h}, t_{0+3h} $= max(CA(t_{0+1h}), CA(t_{0+2h}), CA(t_{0+3h}))$

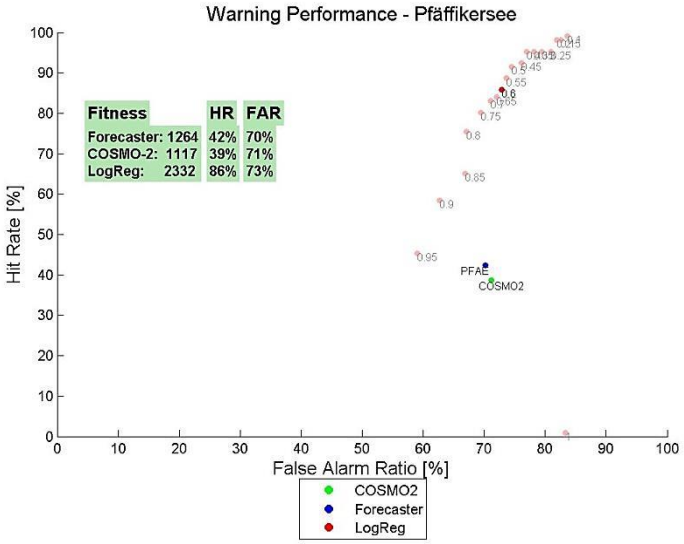
T500, T850	**inst**	Maximum of temperature difference between 850hPa and 500hPa at time t_{0+1h}, t_{0+2h}, t_{0+3h} *= max(T850(t_{0+1h})-T500(t_{0+1h}), T850(t_{0+2h})-T500(t_{0+2h}), T850(t_{0+3h})-T500(t_{0+3h})*
	tdif50	Temperature difference between time t_{0+3h} and t_{0+1h} auf 500 hPa *= T500(t_{0+3h})-T500(t_{0+1h})*
	tdif85	Temperature difference between time t_{0+3h} and t_{0+1h} auf 850 hPa *= T850(t_{0+3h})-T850(t_{0+1h})*
BRN	**brn**	Maximum of the integral of the Bulk Richardson Number between 200 m and 2000 m over ground at time t_{0+1h}, t_{0+2h}, t_{0+3h} *= max(BRN(t_{0+1h}),BRN(t_{0+2h}),BRN(t_{0+3h}))*
PT, PTH	**ptdifmo**	Minimum of the difference of potential temperature between the altitude station HIGH and reference station BAS1 at time t_{0+1h}, t_{0+2h}, t_{0+3h} *= min(PTH(t_{0+1h})-PT(t_{0+1h}), PTH(t_{0+2h})-PT(t_{0+2h}), PTH(t_{0+3h})-PT(t_{0+3h})*
VVH	**mmoh**	Maximum of VMAX_10M forecasts *** 1.94** (in kt) at time t_{0+1h}, t_{0+2h}, t_{0+3h} *= max(VVH(t_{0+1h}), VVH(t_{0+2h}), VVH(t_{0+3h}))*1.94*
TT, T100	**stab**	Difference of temperature between 2 Meter and 100 Meter above ground at time t_{0+1h} *= TT(t_{0+1h})-T100(t_{0+1h})*
DBZ	**dbz**	Maximum of DBZ forecast at time t_{0+1h}, t_{0+2h}, t_{0+3h} *= max(DBZ(t_{0+1h}),DBZ(t_{0+2h}),DBZ(t_{0+3h}))* *With if DBZ < 0 then DBZ = 0*

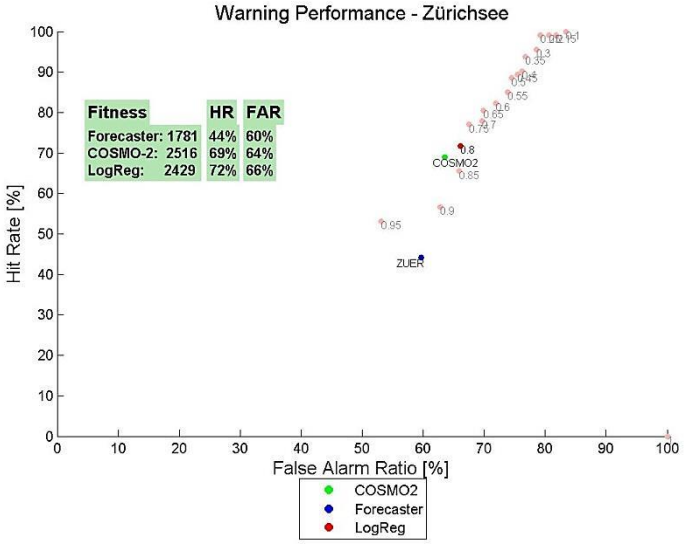
B Appendix: Detailed Warning Performance

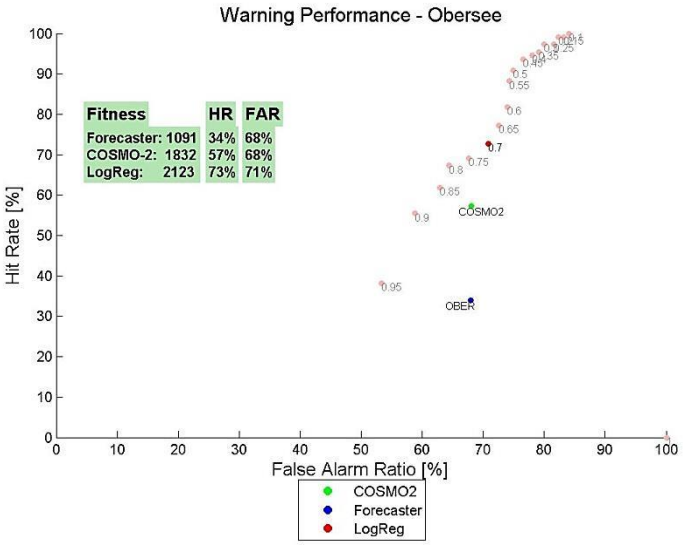


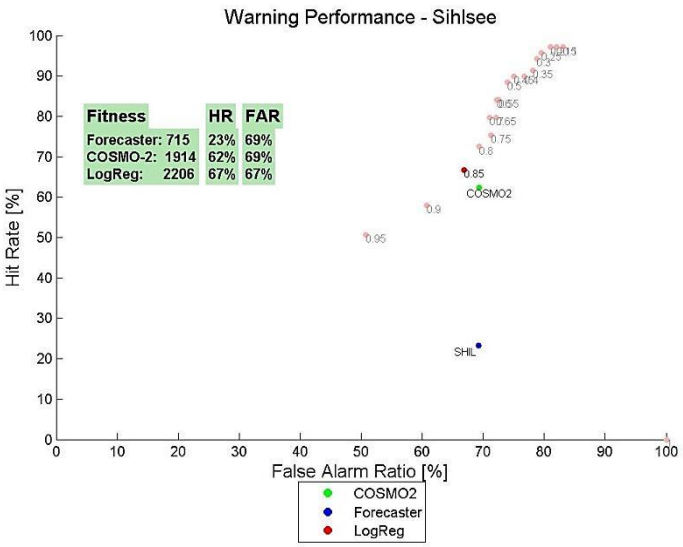


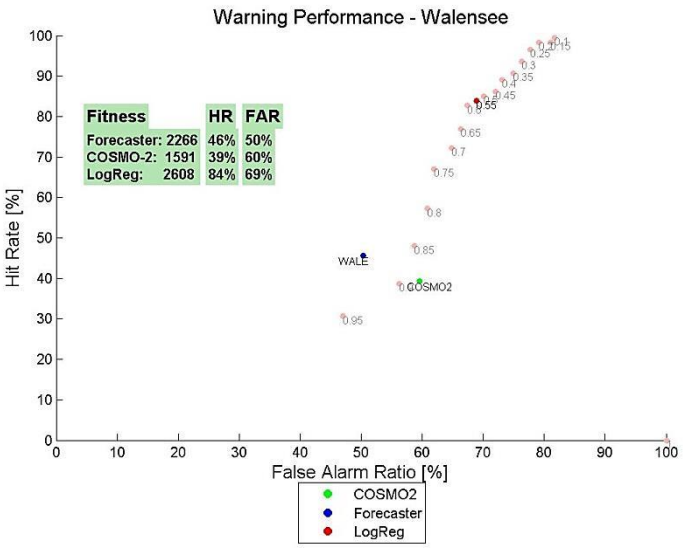

Warning Performance - Bodensee West
Fitness HR FAR
Forecaster: 1375 29% 53%
COSMO-2: 2487 61% 59%
LogReg: 3191 83% 62%
COSMO2
BWES
Hit Rate [%]
False Alarm Ratio [%]
COSMO2
Forecaster
LogReg

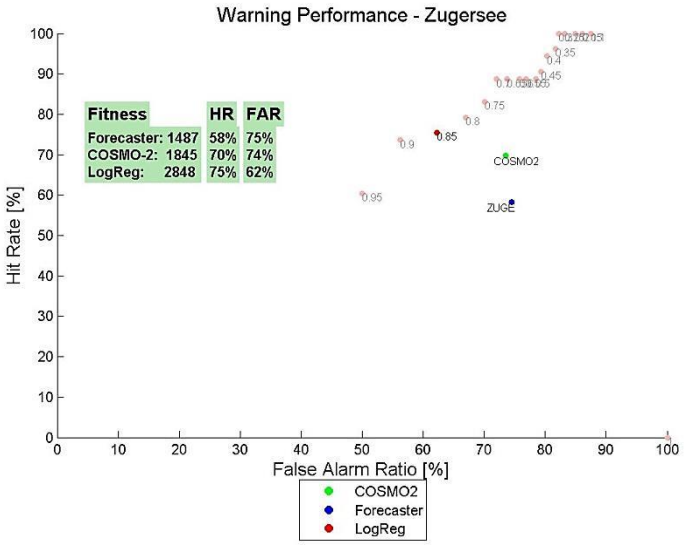

Warning Performance - Pfäffikersee
Fitness HR FAR
Forecaster: 1264 42% 70%
COSMO-2: 1117 39% 71%
LogReg: 2332 86% 73%
PFAE
COSMO2
Hit Rate [%]
False Alarm Ratio [%]
COSMO2
Forecaster
LogReg

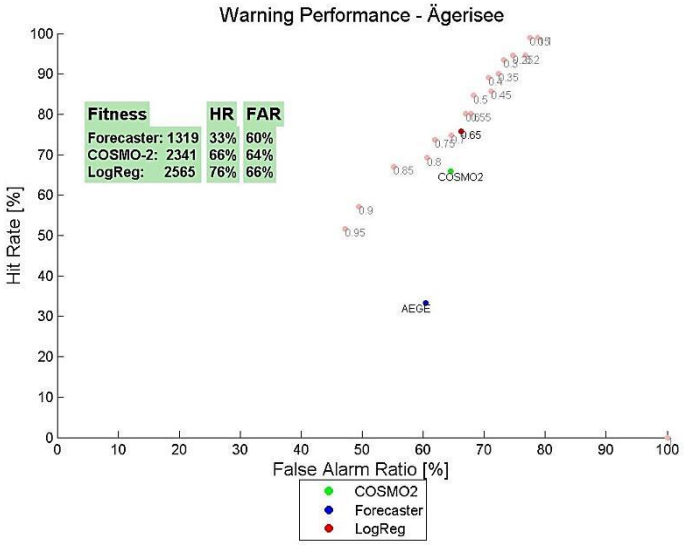

Warning Performance - Zürichsee
Fitness HR FAR
Forecaster: 1781 44% 60%
COSMO-2: 2516 69% 64%
LogReg: 2429 72% 66%
COSMO2
ZUER
Hit Rate [%]
False Alarm Ratio [%]
COSMO2
Forecaster
LogReg

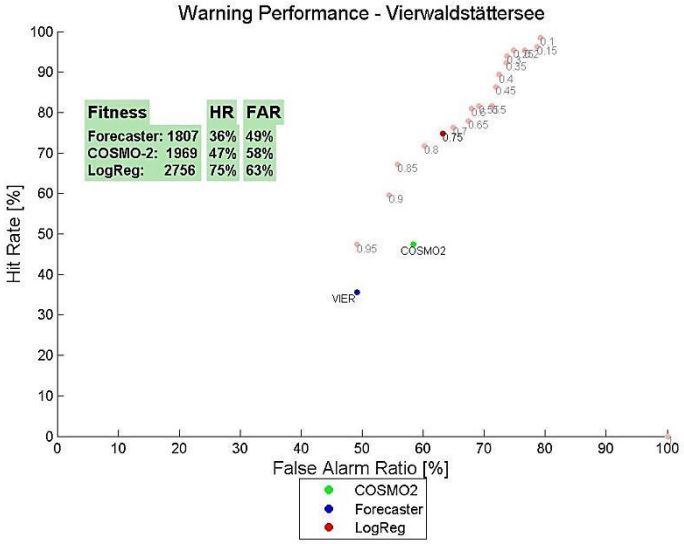

Warning Performance - Obersee
Fitness HR FAR
Forecaster: 1091 34% 68%
COSMO-2: 1832 57% 68%
LogReg: 2123 73% 71%
COSMO2
OBER
Hit Rate [%]
False Alarm Ratio [%]
COSMO2
Forecaster
LogReg

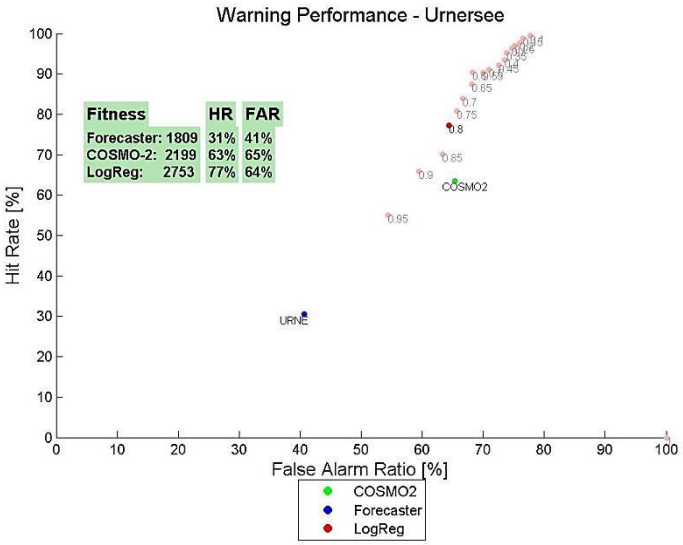

Warning Performance - Sihlsee
Fitness HR FAR
Forecaster: 715 23% 69%
COSMO-2: 1914 62% 69%
LogReg: 2206 67% 67%
COSMO2
SHIL
Hit Rate [%]
False Alarm Ratio [%]
COSMO2
Forecaster
LogReg

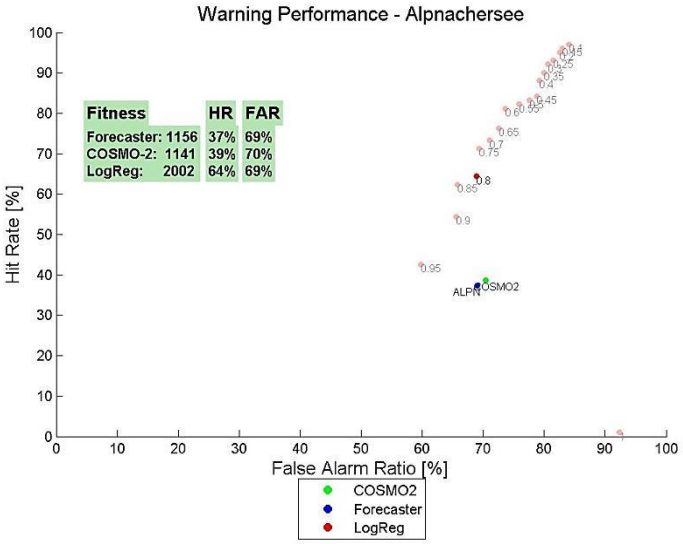

Warning Performance - Walensee
Fitness HR FAR
Forecaster: 2266 46% 50%
COSMO-2: 1591 39% 60%
LogReg: 2608 84% 69%
WALE
COSMO2
Hit Rate [%]
False Alarm Ratio [%]
COSMO2
Forecaster
LogReg

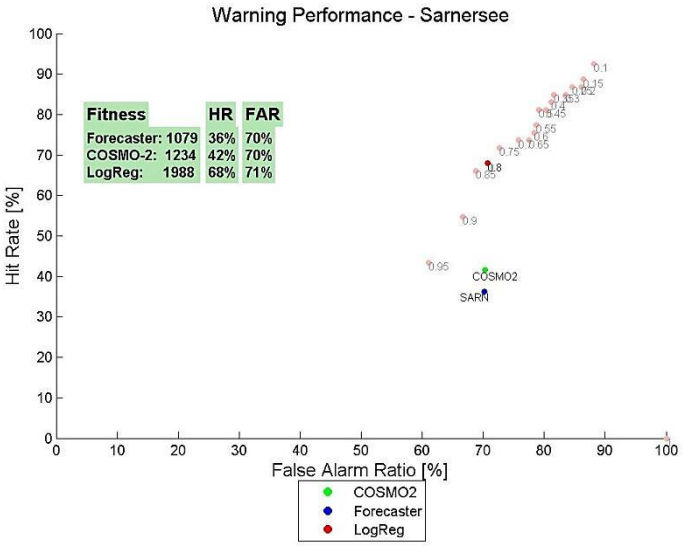

Warning Performance - Zugersee
Fitness HR FAR
Forecaster: 1487 58% 75%
COSMO-2: 1845 70% 74%
LogReg: 2848 75% 62%
ZUGE
COSMO2
Hit Rate [%]
False Alarm Ratio [%]
COSMO2
Forecaster
LogReg

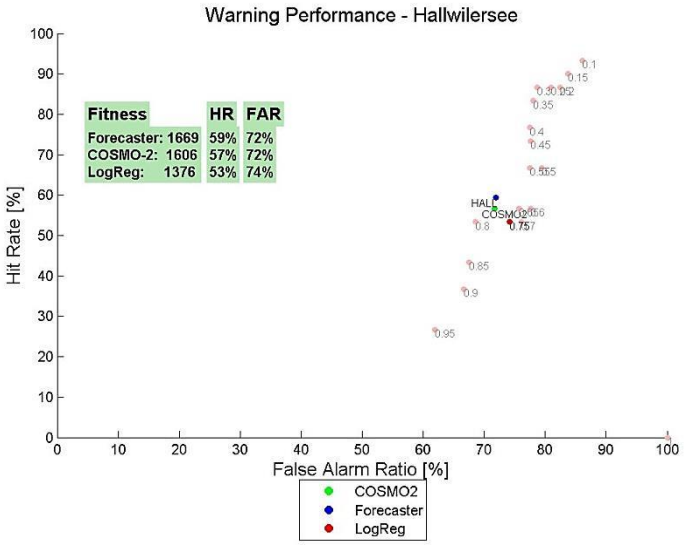

Warning Performance - Ägerisee
Fitness HR FAR
Forecaster: 1319 33% 60%
COSMO-2: 2341 66% 64%
LogReg: 2565 76% 66%
AEGE
COSMO2
Hit Rate [%]
False Alarm Ratio [%]
COSMO2
Forecaster
LogReg

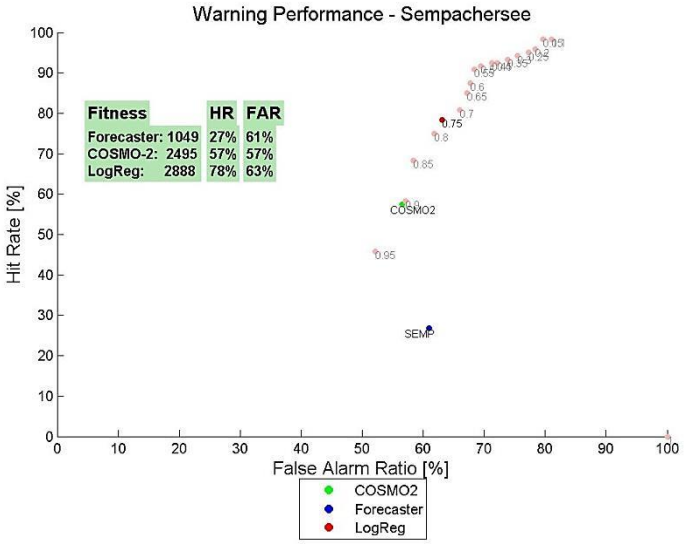

Warning Performance - Vierwaldstättersee
Fitness HR FAR
Forecaster: 1807 36% 49%
COSMO-2: 1969 47% 58%
LogReg: 2756 75% 63%
VIER
COSMO2
Hit Rate [%]
False Alarm Ratio [%]
COSMO2
Forecaster
LogReg

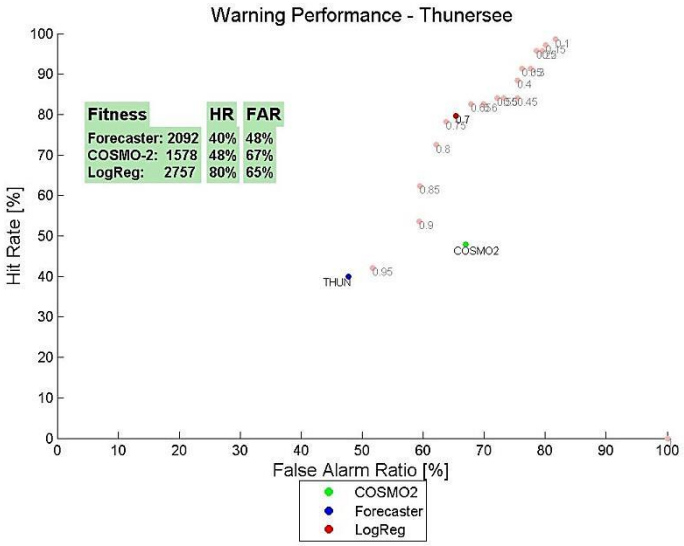

Warning Performance - Urnersee
Fitness HR FAR
Forecaster: 1809 31% 41%
COSMO-2: 2199 63% 65%
LogReg: 2753 77% 64%
Hit Rate [%]
False Alarm Ratio [%]
COSMO2
URNE
COSMO2
Forecaster
LogReg

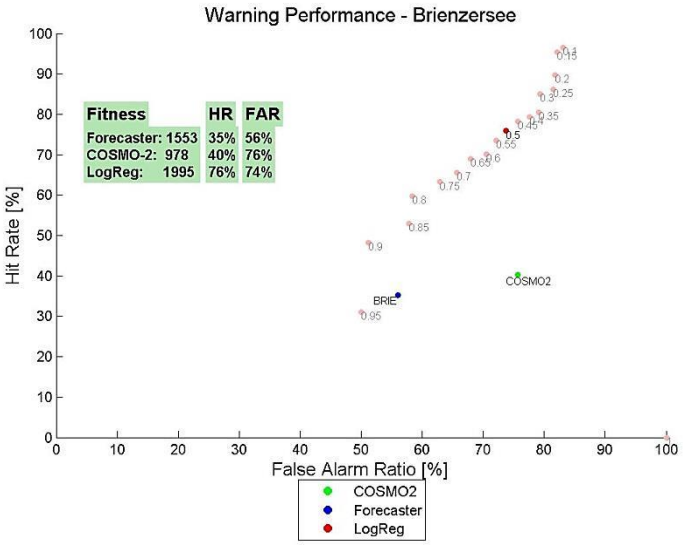

Warning Performance - Alpnachersee
Fitness HR FAR
Forecaster: 1156 37% 69%
COSMO-2: 1141 39% 70%
LogReg: 2002 64% 69%
Hit Rate [%]
False Alarm Ratio [%]
ALPN
COSMO2
COSMO2
Forecaster
LogReg

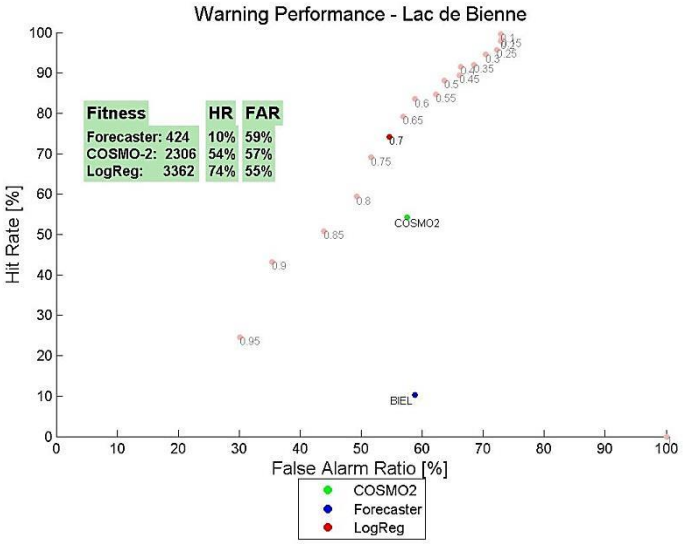

Warning Performance - Sarnersee
Fitness HR FAR
Forecaster: 1079 36% 70%
COSMO-2: 1234 42% 70%
LogReg: 1988 68% 71%
Hit Rate [%]
False Alarm Ratio [%]
COSMO2
SARN
COSMO2
Forecaster
LogReg

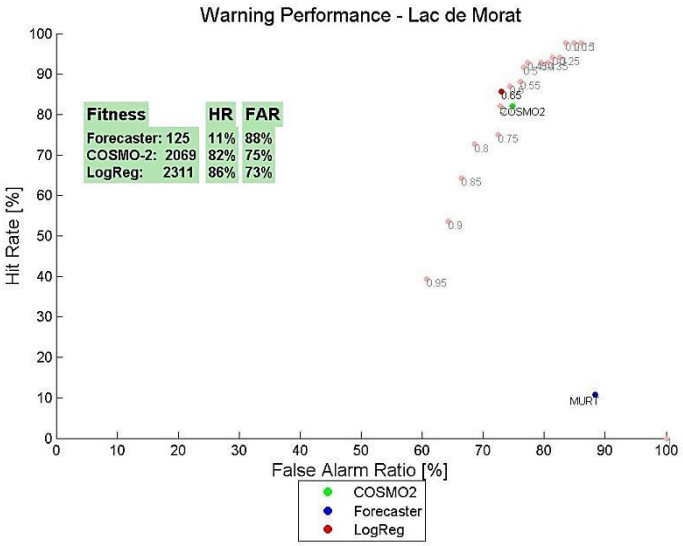

Warning Performance - Hallwilersee
Fitness HR FAR
Forecaster: 1669 59% 72%
COSMO-2: 1606 57% 72%
LogReg: 1376 53% 74%
HAL
COSMO2
Hit Rate [%]
False Alarm Ratio [%]
COSMO2
Forecaster
LogReg

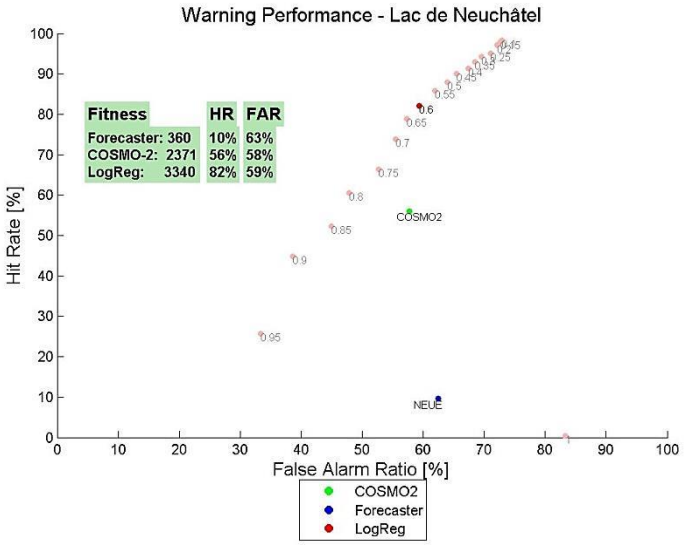

Warning Performance - Sempachersee
Fitness HR FAR
Forecaster: 1049 27% 61%
COSMO-2: 2495 57% 57%
LogReg: 2888 78% 63%
COSMO2
SEMP
Hit Rate [%]
False Alarm Ratio [%]
COSMO2
Forecaster
LogReg

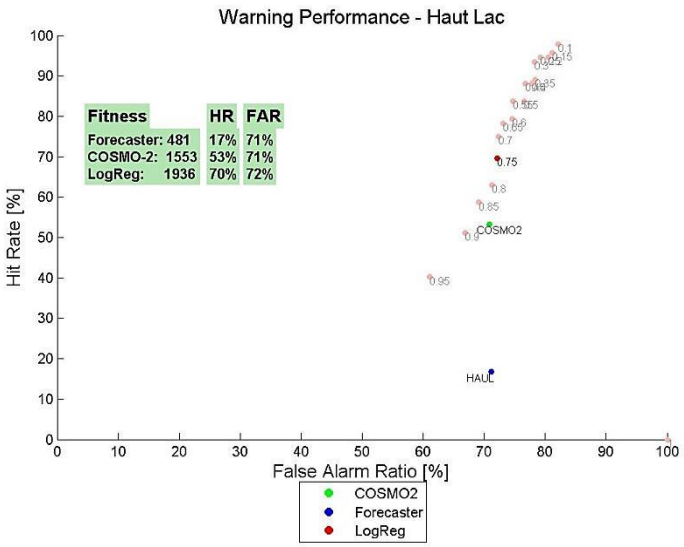

Warning Performance - Thunersee
Fitness HR FAR
Forecaster: 2092 40% 48%
COSMO-2: 1578 48% 67%
LogReg: 2757 80% 65%
COSMO2
THUN
Hit Rate [%]
False Alarm Ratio [%]
COSMO2
Forecaster
LogReg

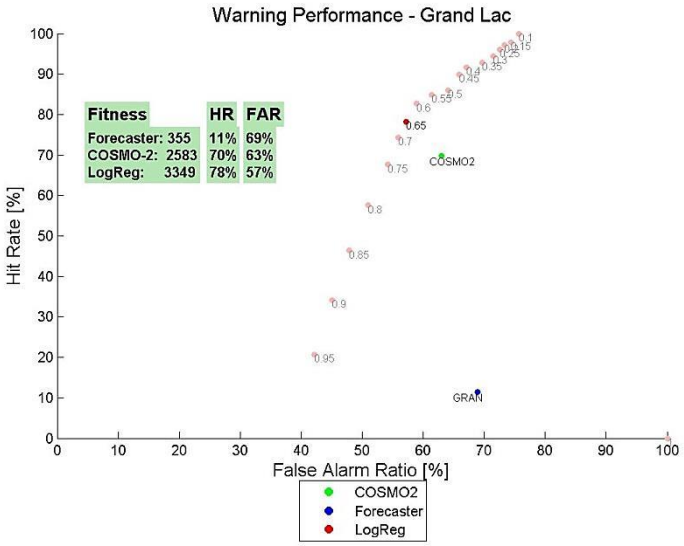

Warning Performance - Brienzersee
Fitness HR FAR
Forecaster: 1553 35% 56%
COSMO-2: 978 40% 76%
LogReg: 1995 76% 74%
Hit Rate [%]
False Alarm Ratio [%]
COSMO2
BRIE
COSMO2
Forecaster
LogReg

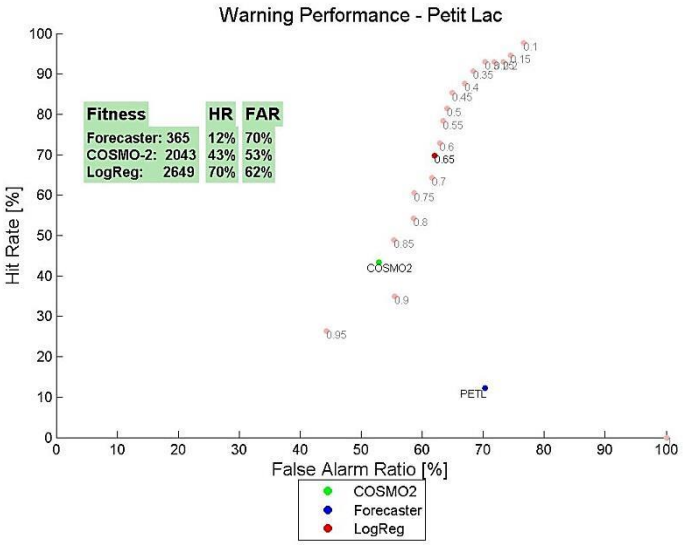

Warning Performance - Lac de Bienne
Fitness HR FAR
Forecaster: 424 10% 59%
COSMO-2: 2306 54% 57%
LogReg: 3362 74% 55%
Hit Rate [%]
False Alarm Ratio [%]
COSMO2
BIEL
COSMO2
Forecaster
LogReg

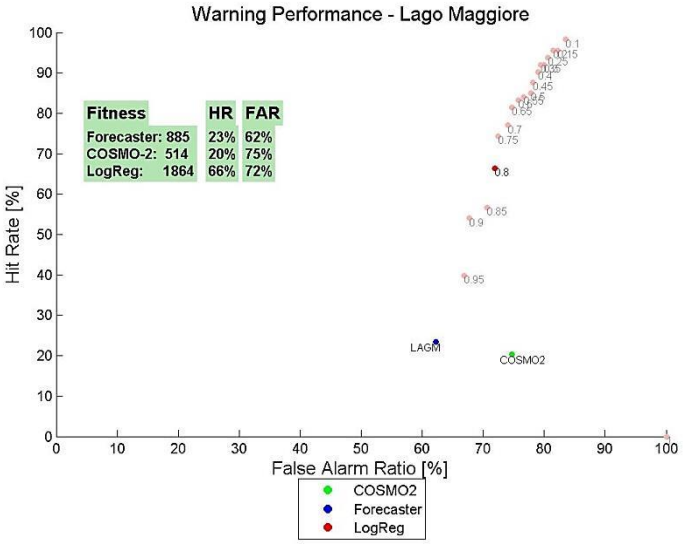

Warning Performance - Lac de Morat
Fitness HR FAR
Forecaster: 125 11% 88%
COSMO-2: 2069 82% 75%
LogReg: 2311 86% 73%
Hit Rate [%]
False Alarm Ratio [%]
COSMO2
MURT
COSMO2
Forecaster
LogReg

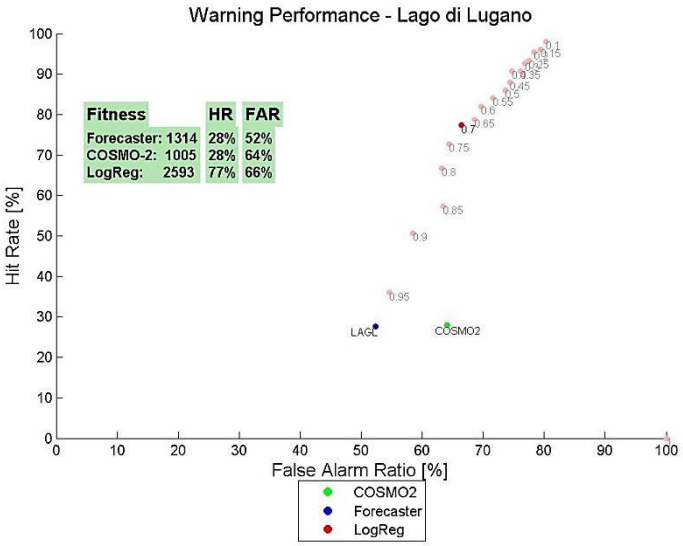

Warning Performance - Lac de Neuchâtel
Fitness HR FAR
Forecaster: 360 10% 63%
COSMO-2: 2371 56% 58%
LogReg: 3340 82% 59%
COSMO2
NEUE
Hit Rate [%]
False Alarm Ratio [%]
COSMO2
Forecaster
LogReg

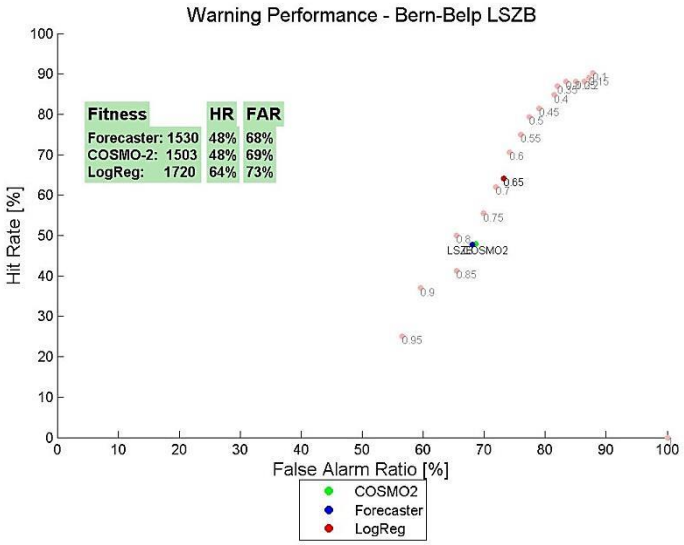

Warning Performance - Haut Lac
Fitness HR FAR
Forecaster: 481 17% 71%
COSMO-2: 1553 53% 71%
LogReg: 1936 70% 72%
COSMO2
HAUL
Hit Rate [%]
False Alarm Ratio [%]
COSMO2
Forecaster
LogReg

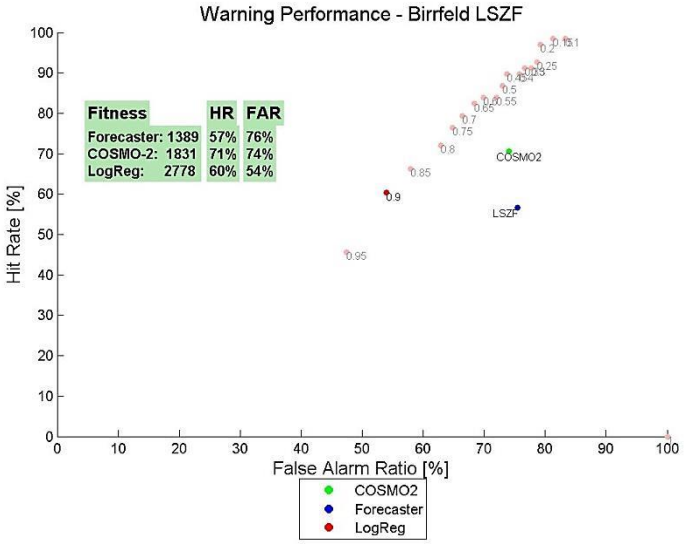

Warning Performance - Grand Lac
Fitness HR FAR
Forecaster: 355 11% 69%
COSMO-2: 2583 70% 63%
LogReg: 3349 78% 57%
COSMO2
GRAN
Hit Rate [%]
False Alarm Ratio [%]
COSMO2
Forecaster
LogReg

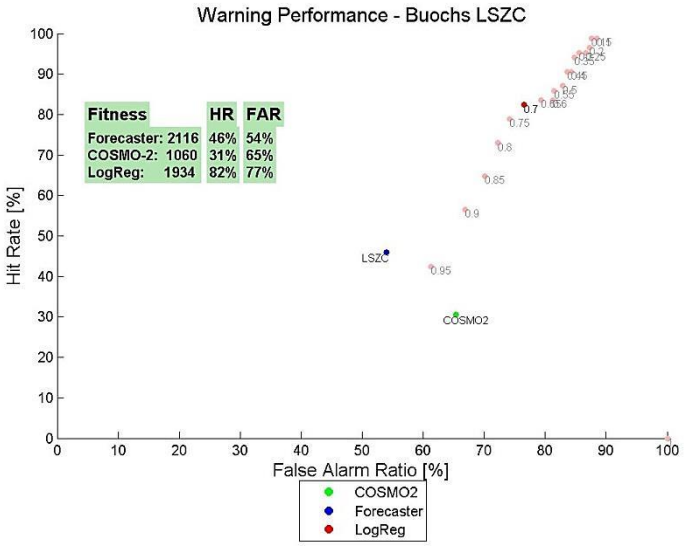

Warning Performance - Petit Lac
Fitness HR FAR
Forecaster: 365 12% 70%
COSMO-2: 2043 43% 53%
LogReg: 2649 70% 62%
Hit Rate [%]
False Alarm Ratio [%]
COSMO2
PETL
COSMO2
Forecaster
LogReg

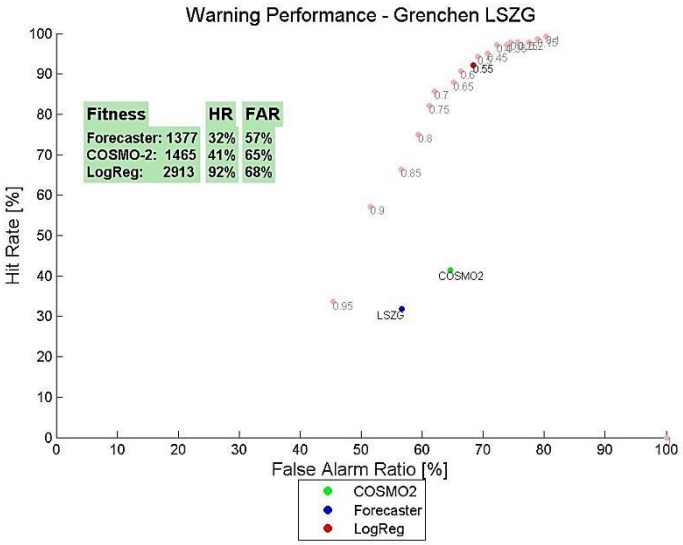

Warning Performance - Lago Maggiore
Fitness HR FAR
Forecaster: 885 23% 62%
COSMO-2: 514 20% 75%
LogReg: 1864 66% 72%
Hit Rate [%]
False Alarm Ratio [%]
LAGM
COSMO2
COSMO2
Forecaster
LogReg

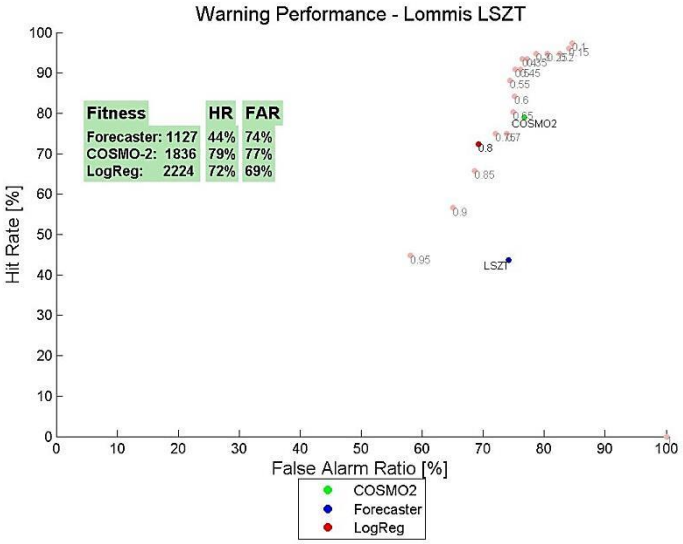

Warning Performance - Lago di Lugano
Fitness HR FAR
Forecaster: 1314 28% 52%
COSMO-2: 1005 28% 64%
LogReg: 2593 77% 66%
Hit Rate [%]
False Alarm Ratio [%]
LAGL
COSMO2
COSMO2
Forecaster
LogReg

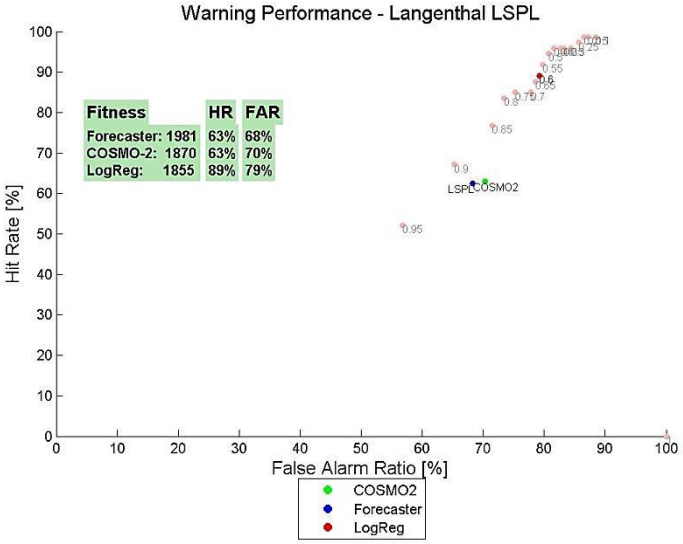

Warning Performance - Bern-Belp LSZB
Fitness HR FAR
Forecaster: 1530 48% 68%
COSMO-2: 1503 48% 69%
LogReg: 1720 64% 73%
Hit Rate [%]
False Alarm Ratio [%]
COSMO2
Forecaster
LogReg

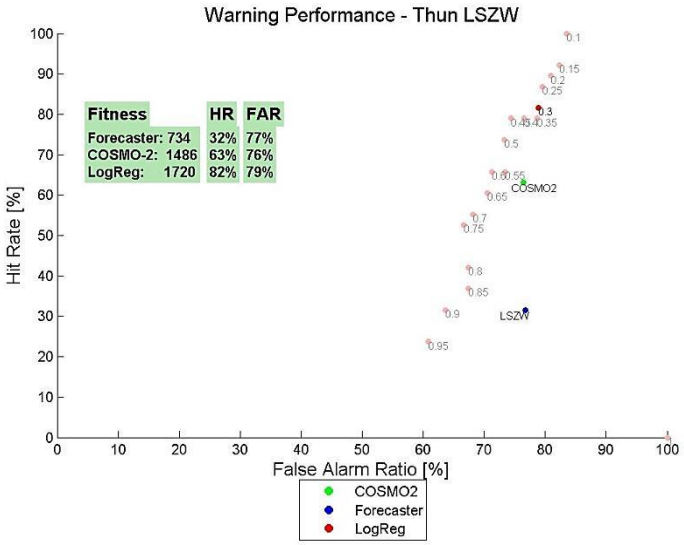

Warning Performance - Birrfeld LSZF
Fitness HR FAR
Forecaster: 1389 57% 76%
COSMO-2: 1831 71% 74%
LogReg: 2778 60% 54%
Hit Rate [%]
False Alarm Ratio [%]
COSMO2
Forecaster
LogReg


Warning Performance - Buochs LSZC
Fitness HR FAR
Forecaster: 2116 46% 54%
COSMO-2: 1060 31% 65%
LogReg: 1934 82% 77%
Hit Rate [%]
False Alarm Ratio [%]
COSMO2
Forecaster
LogReg

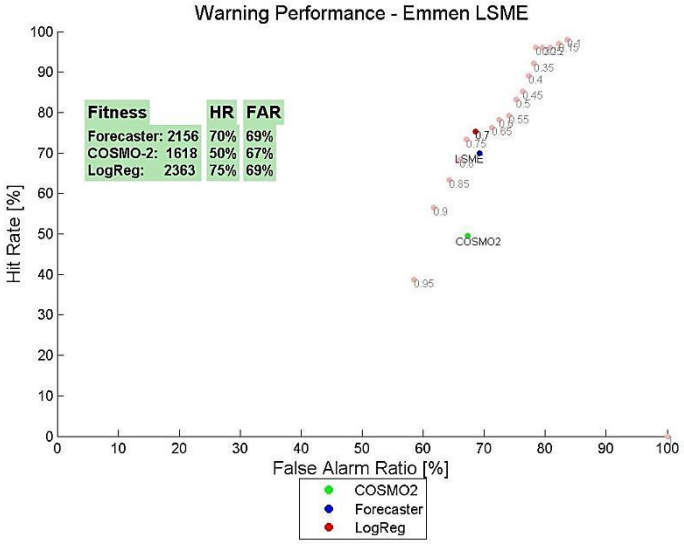

Warning Performance - Grenchen LSZG
Fitness HR FAR
Forecaster: 1377 32% 57%
COSMO-2: 1465 41% 65%
LogReg: 2913 92% 68%
Hit Rate [%]
False Alarm Ratio [%]
COSMO2
LSZG
COSMO2
Forecaster
LogReg

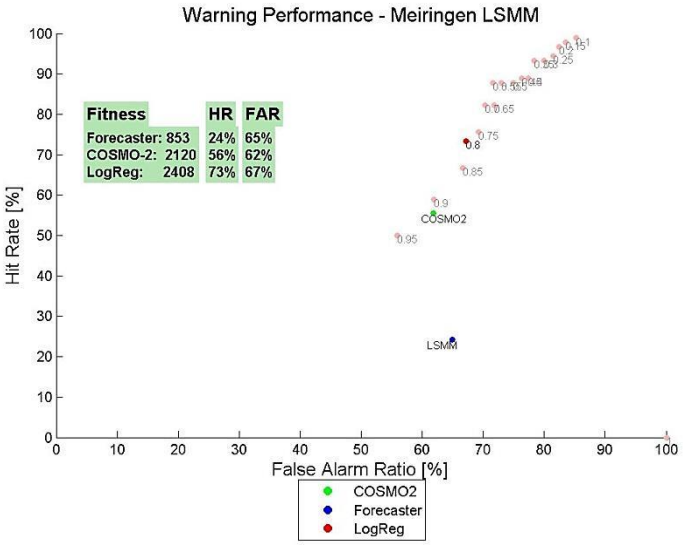

Warning Performance - Lommis LSZT
Fitness HR FAR
Forecaster: 1127 44% 74%
COSMO-2: 1836 79% 77%
LogReg: 2224 72% 69%
Hit Rate [%]
False Alarm Ratio [%]
COSMO2
LSZT
COSMO2
Forecaster
LogReg

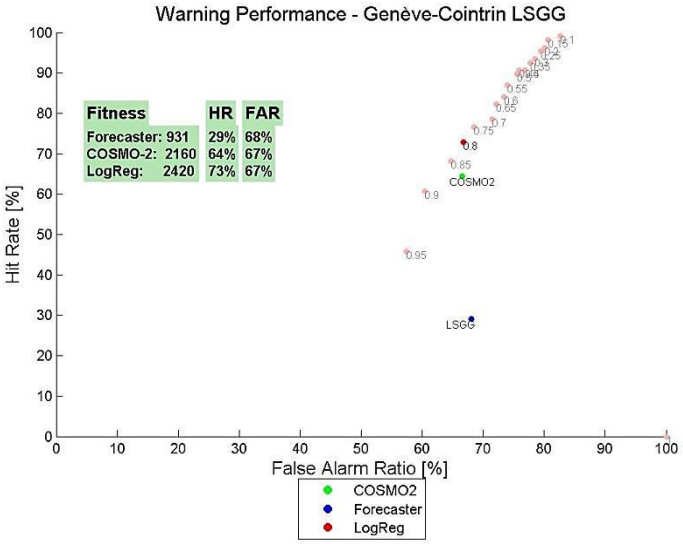

Warning Performance - Langenthal LSPL
Fitness HR FAR
Forecaster: 1981 63% 68%
COSMO-2: 1870 63% 70%
LogReg: 1855 89% 79%
Hit Rate [%]
False Alarm Ratio [%]
LSPL
COSMO2
COSMO2
Forecaster
LogReg

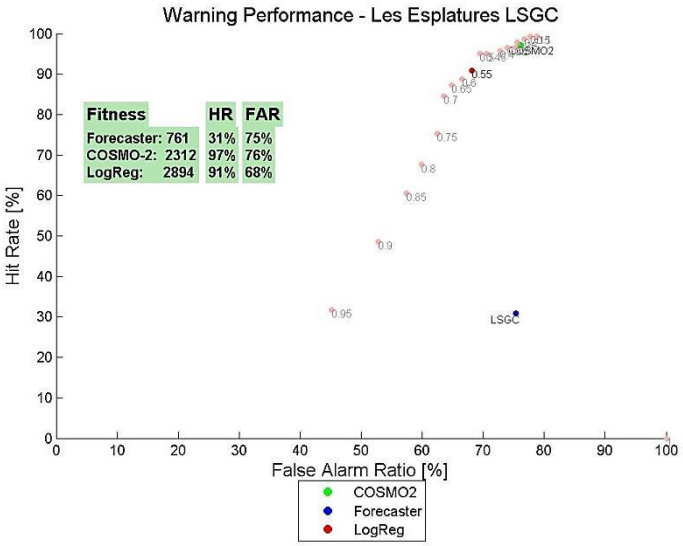

Warning Performance - Thun LSZW
Fitness HR FAR
Forecaster: 734 32% 77%
COSMO-2: 1486 63% 76%
LogReg: 1720 82% 79%
Hit Rate [%]
False Alarm Ratio [%]
COSMO2
LSZW
COSMO2
Forecaster
LogReg

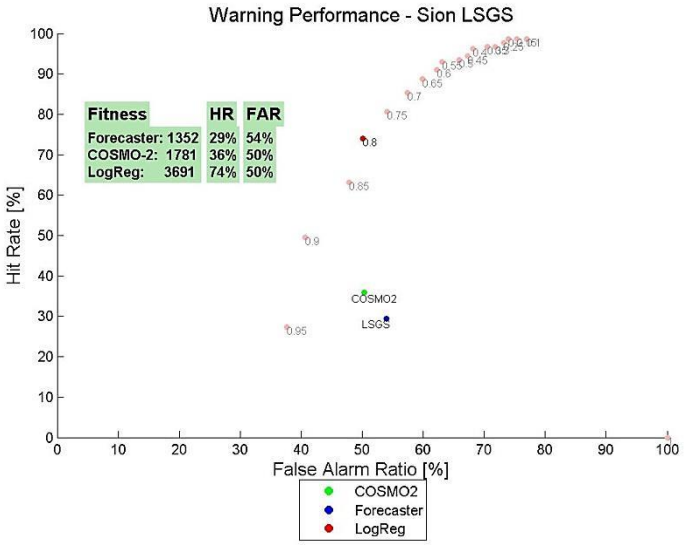

Warning Performance - Zürich-Kloten LSZH
Fitness HR FAR
Forecaster: 1977 41% 51%
COSMO-2: 1517 37% 59%
LogReg: 3072 84% 64%
Hit Rate [%]
False Alarm Ratio [%]
LSZH
COSMO2
COSMO2
Forecaster
LogReg

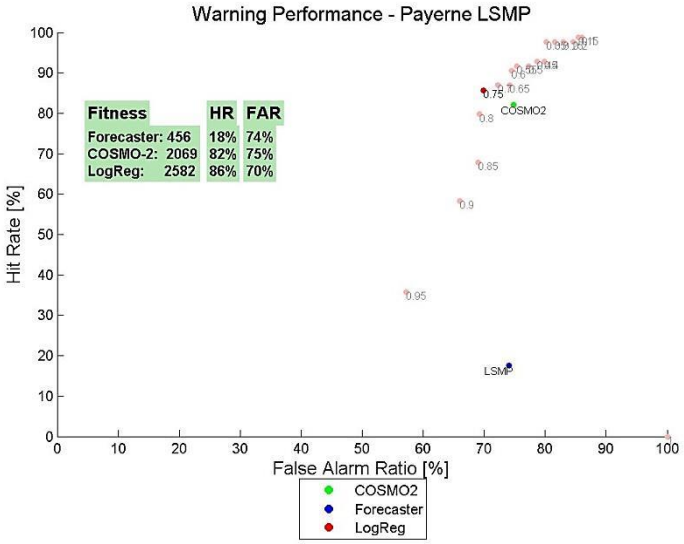

Warning Performance - Emmen LSME
Fitness HR FAR
Forecaster: 2156 70% 69%
COSMO-2: 1618 50% 67%
LogReg: 2363 75% 69%
Hit Rate [%]
False Alarm Ratio [%]
LSME
COSMO2
COSMO2
Forecaster
LogReg

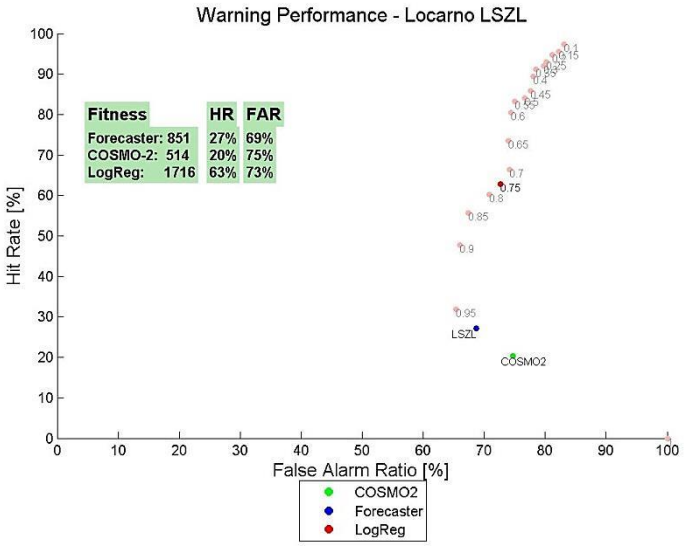

Warning Performance - Meiringen LSMM
Fitness HR FAR
Forecaster: 853 24% 65%
COSMO-2: 2120 56% 62%
LogReg: 2408 73% 67%
COSMO2
LSMM
Hit Rate [%]
False Alarm Ratio [%]
COSMO2
Forecaster
LogReg


Warning Performance - Genève-Cointrin LSGG
Fitness HR FAR
Forecaster: 931 29% 68%
COSMO-2: 2160 64% 67%
LogReg: 2420 73% 67%
COSMO2
LSGG
Hit Rate [%]
False Alarm Ratio [%]
COSMO2
Forecaster
LogReg


Warning Performance - Les Esplatures LSGC
Fitness HR FAR
Forecaster: 761 31% 75%
COSMO-2: 2312 97% 76%
LogReg: 2894 91% 68%
LSGC
Hit Rate [%]
False Alarm Ratio [%]
COSMO2
Forecaster
LogReg


Warning Performance - Sion LSGS
Fitness HR FAR
Forecaster: 1352 29% 54%
COSMO-2: 1781 36% 50%
LogReg: 3691 74% 50%
COSMO2
LSGS
Hit Rate [%]
False Alarm Ratio [%]
COSMO2
Forecaster
LogReg


Warning Performance - Payerne LSMP
Fitness HR FAR
Forecaster: 456 18% 74%
COSMO-2: 2069 82% 75%
LogReg: 2582 86% 70%
COSMO2
LSMP
Hit Rate [%]
False Alarm Ratio [%]
COSMO2
Forecaster
LogReg


Warning Performance - Locarno LSZL
Fitness HR FAR
Forecaster: 851 27% 69%
COSMO-2: 514 20% 75%
LogReg: 1716 63% 73%
LSZL
COSMO2
Hit Rate [%]
False Alarm Ratio [%]
COSMO2
Forecaster
LogReg

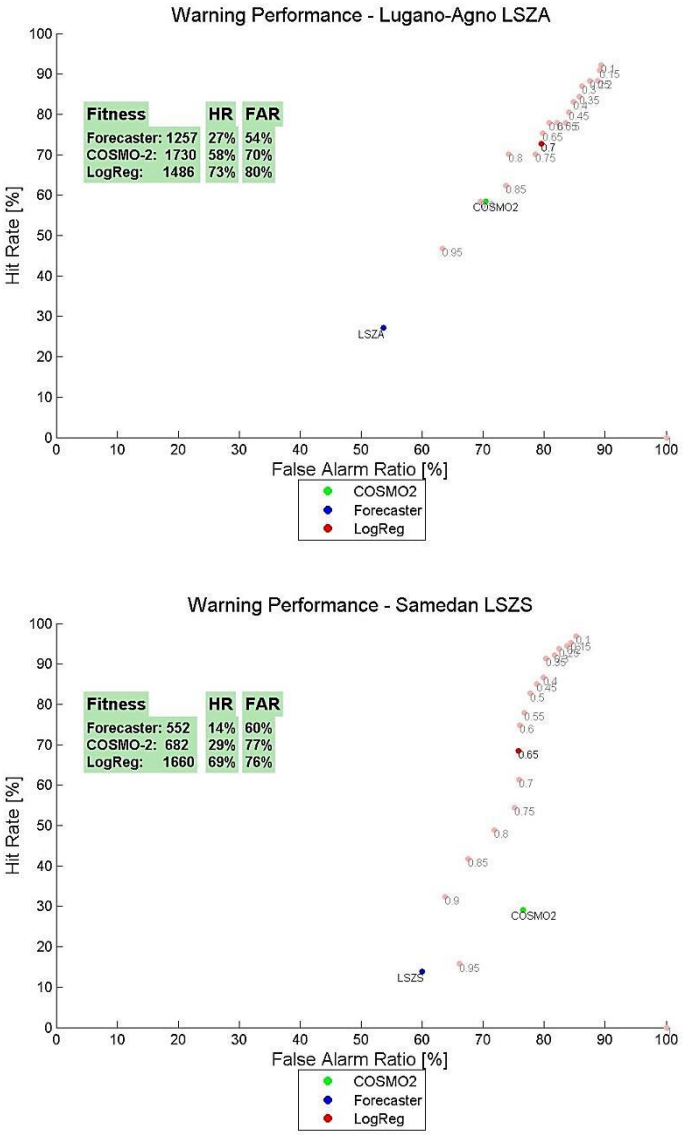


Figure 26: The plots represent the FAR on the x-axis and HR on the y-axis . Red points: results of the verification conducted on the 2-year independent data set of the GaleWarn system for the specified warning object. The probability of occurrence is written next to each point. The optimal probability threshold determined on the learn-data is displayed in bright red colour. Green point: result of the verification on the 2-year independent data set of the COSMO-2 direct model output (parameter maximal wind gust). Blue point: result of the AWV1.0 basis verification of the manual warnings issued for the specified object over the years 2011 to 2013. The fitness values of each warning type are written on each diagram.

MeteoSchweiz
Operation Center 1
CH-8044 Zürich-Flughafen
T +41 58 460 99 99
www.meteoschweiz.ch

MeteoSvizzera
Via ai Monti 146
CH-6605 Locarno Monti
T +41 58 460 97 77
www.meteosvizzera.ch

MétéoSuisse
7bis, av. de la Paix
CH-1211 Genève 2
T +41 58 460 98 88
www.meteosuisse.ch

MétéoSuisse
Chemin de l'Aérologie
CH-1530 Payerne
T +41 58 460 94 44
www.meteosuisse.ch