Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 238 (2024) 49–56

Procedia Computer Science

www.elsevier.com/locate/procedia

The 15th International Conference on Ambient Systems, Networks and Technologies April 23-25, 2024, Hasselt, Belgium

# An implementation, evaluation and validation of a dynamic fire and conflagration risk indicator for wooden homes

R.D. Strand[a,*], L.M. Kristensen[b]

[a]*Department of Safety, Chemistry, and Biomedical laboratory sciences, Western Norway University of Applied Sciences, Haugesund, Norway*
[b]*Department of Computer Science, Electrical Engineering and Mathematical Sciences, Western Norway University of Applied Sciences, Bergen, Norway*

**Abstract**

The pervasive presence of IoT sensing devices combined with advances in cloud-based services has made historical- and forecast weather data services widely available for use in smart software systems. The contribution of this paper is the design, performance evaluation, and validation of a software implementation of model-based dynamic fire risk prediction for wooden homes using local weather data. A key feature of the implementation is that the software architecture has been designed to support the use of different underlying cloud-based weather data services, and the integration as a service in third-party smart systems embedding fire risk predictions. The performed evaluation shows that the implementation is efficient, as weather data can be retrieved, preprocessed and fire risk predictions computed within seconds using only in the order of Kb's of memory; and accurate considering our sets of data, as the fire risks computed by the implementation have been validated against a set of in-situ measurements.



*Keywords:* Climate-related risks; Mitigation of urban fire risks; Reducing vulnerability in cities; Smart software systems and services

## 1. Introduction

Dynamic risk assessments have typically been associated with the process- and nuclear industry, where multiple safety barriers or safeguards are implemented to monitor and control processes and to prevent and mitigate undesired events. In such systems, the risk levels or plant status can rely on the status of the safety barriers, e.g., considered through condition monitoring [16]. Still, dynamic risk assessment has long been applied to systems without complex safety barriers. For these systems, the imminent risk is based on the status of the system itself, like forest fire risk and the associated forest fire indices, where models typically estimate fuel conditions like the fuel moisture content

* Corresponding author.
*E-mail address:* rds@hvl.no


10.1016/j.procs.2024.05.195

(FMC) based on the occurring and predicted weather. Similarly, but less known, the in-home fire risk of wooden homes may also be indicated for the present and near future through risk modelling based on weather data such as outdoor temperature, relative humidity (RH) and wind speed [10, 14, 15].

Regardless of the approach, a modelled status of a system relies on available data. In recent years, increasing amounts of data have been collected through a growing network of Internet of Things (IoT) appliances, supporting the harvesting of location-specific data. Combined with the pervasive presence of cloud-based data services, it has provided access to a wide range of data sources and enabled the development of data-driven applications for dynamic risk assessments. A prominent example of IoT appliances and publicly available cloud-based services is the REST API weather data service provided by the Norwegian Meteorological Institute (MET) [9], where the weather forecast is improved by using measurements from consumer-grade weather stations belonging to the Netatmo network [7].

Weather data is widely used in many sectors and industries, e.g., aviation, agriculture and energy. It is also well known that forest fire risk heavily relies on the occurring weather, making it a seasonal risk. Forest fire danger indices typically estimate fuel conditions, where models are applied to large geographical areas with similar vegetation despite local variations. The in-home fire risk of wooden homes inherits the forest fire risk modelling principle when estimating fire risk for homes with internal wooden claddings, especially in the ceilings and walls. Recognising the effect of in-home FMC of wooden panels on enclosure fire development enables the computation of a wooden home fire risk (WHFR) indicator. Combined with wind forecasts, this may produce a conflagration indicator for dense wooden home areas, e.g., wooden house heritage sites, where a parallel can be drawn to densely situated trees in a forest. The WHFR model estimates in-home RH and FMC of wooden panels using outdoor weather data. It is well known that fire in dry fuel may develop faster and burn more fiercely. In terms of enclosure fires, this means reaching flashover at an earlier stage compared to wood at higher humidities. This leaves less time for safe escape and for the fire brigades to respond to an event before critical conditions occur.

Earlier and related work has validated the fire risk model [15, 12] and justified the modelling approach [4] that provides the foundation of this paper. In addition, the model has been evaluated on several historical fires in Norway [3] and shown its practicability [4]. The WHFR indicator, i.e., the FMC, has been correlated with the time to flashover (TTF) according to [1] in an attempt to produce an intuitive fire risk indicator. In particular, a mobile application using edge computing implemented the model and communicated computed risks through a graphical user interface (GUI) [2]. In [4], the modelled TTF was indexed according to observations from Norwegian conditions, resulting in a proposed wooden home fire danger index. The developed model and GUI, still subject to evaluation and validation, may eventually become a wooden home and conflagration warning system supporting decision-making and control in cold climate regions, e.g., Scandinavia, Finland, Canada and possibly countries in climatic regions like Japan and New Zealand. Considering wooden home conflagration risk, densely built wooden home areas, such as the Norwegian wooden home heritage sites, are especially prone. Dry indoor conditions and strong winds are known conflagration factors [5]. In the present context, wooden homes refer to homes with a certain fraction of indoor wooden panels [4].

This paper focuses on the design and evaluation of a Python implementation of the WHFR model underlying our earlier work. The implementation is designed to be agnostic to the cloud-based weather data sources and to be integrated into third-party systems. We evaluate the implementation performance in terms of computation time and memory use. We verify and validate the model output, in terms of the predicted RH and TTF, against baseline modelling results and sets of measured data (indoor RH) from the Haugesund region in Norway. The full implementation is available via [13].

**Outline.** In Sec. 2, the main elements of the fire risk model are presented, while in Sec. 3 the software architecture and data model are introduced. Section 4 and 5 present the fire risk computation service and the data harvesting service, responsible for fire risk computations and weather data harvesting, respectively. The implementation and fire risk model is evaluated in Sec. 6. Finally, in Sec. 7, our main findings are summarised.

## 2. Fire Risk Model

This section outlines the main elements of the implemented wooden home fire risk model. For full details, readers are referred to [15]. The model uses outdoor RH and outdoor temperature to compute indoor RH and indoor FMC of wooden surfaces from first principle mathematics and physics. Changes in indoor water concentration are considered a function of (1) air changes due to ventilation and leaks; (2) internal water supply from, e.g., cooking, respiring, and plants; and (3) humidity exchange between bulk air and hygroscopic surfaces, i.e., wooden surfaces. The main differential equation to solve when implementing the model is [15]:

$$dC/dt = \dot{C}_{ac} + \dot{C}_{supply} + \dot{C}_{surf} \tag{1}$$

where $C$ (kg · m$^{-3}$) is the water vapour concentration of the indoor air, $\dot{C}_{ac}$ (kg/m$^3$ · s) is the change in water vapour concentration through air changes, $\dot{C}_{supply}$ (kg/m$^3$ · s) is the change in water vapour concentration from in-home water vapour production, and $\dot{C}_{surf}$ (kg/m$^3$ · s) is the change in water vapour concentration due to humidity exchange between bulk air and wooden (hygroscopic) surfaces. The water vapour concentration of the bulk air, $C$, is eventually used to compute the indoor RH and, consequently, the FMC of wooden surfaces.

For enclosure-specific parameters, the model uses a generic enclosure representing a combined kitchen and living room as justified in [4]. The air change rate per hour (ACH) is modelled as natural ventilation, where the inflow of ambient air depends on the temperature difference between outdoor and indoor air. The ACH is given by:

$$ACH = \gamma \cdot \sqrt{(1/T_{out} - 1/T_{in})/T_{out}} \tag{2}$$

where $T_{out}$ (K) is outdoor temperature and $T_{in}$ (K) is indoor temperature. The term involving the square root represents the inflow rate according to the Bernoulli principle. The $\gamma$ then becomes a ventilation factor adjusting for proper ventilation rates, where the recommended range is 350 - 400, according to [4]. Considering the in-home humidity supply, the value is determined from literature in combination with modelling efforts [15]. The recommended range is given as 0.8 - 1.2 (Kg/day) with a recommended value of 1 (Kg/day) [4].

The majority of the modelling effort relates to the computation of the wooden surface FMC and the water exchange between the bulk air and these wooden surfaces. The RH in the boundary layer at the wooden surfaces is a linear function of the RH corresponding to the wooden surface FMC (2-3 mm depth) and the RH of the bulk air. The wooden surfaces are considered pine wood of thickness $L$ subdivided into $N$ layers, making the numerical layer thickness equal to $\triangle x = L/N$. Typically, for Norwegian conditions, 12 mm pine wood is subdivided into 10 layers [15]. The water concentration in the layer of wood facing the enclosure is computed as [15]:

$$C_{1_{(t+\Delta t)}} = C_{1_{(t)}} + \frac{\Delta t}{\Delta x} \cdot \left( \frac{D_{w,a}}{\delta} \cdot (RH_{in_{(t)}} - RH_{wall_{(t)}}) \cdot C_{sat,in} + \frac{D_{w,s}}{\Delta x} \cdot (C_{2_{(t)}} - C_{1_{(t)}}) \right) \tag{3}$$

where $C_{1_{(t+\Delta t)}}$ (Kg/m$^3$) is the water concentration for layer $n = 1$ at the next timestep, $\triangle t$ (s) is the time step, $D_{w,a}$ (m$^2$/s) is the diffusion coefficient of water vapour in air at the specified indoor temperature, $D_{w,s}$ (m$^2$/s) is the solid wood water diffusion coefficient, $\delta$ (m) is the boundary layer thickness, taken to be $0.01m$ [15]. For the layers of wood $n = 2$ to $n = N - 1$, the water concentrations are obtained by solving a second-order partial differential equation.

The modelled FMC can then be correlated with the time to flashover (TTF) according to the curve fit [1]:

$$TTF = 2.0 \cdot exp(0.16 \cdot FMC) \tag{4}$$

where TTF is the final wooden home fire risk indicator, and FMC is the percentage by weight of water in the wood. In general, low indoor RH results in low FMC and low TTF values, corresponding to a high fire risk.

## 3. Software Architecture and Data Model

The overall software architecture of the fire risk model implementation is presented in Fig. 1(left). The software architecture is comprised of four main parts: The FireRiskComputation API (FRC API) provides the services that applications can use when embedding the implementation into a system solution; the FireRiskComputationModel (FRC) module implementing the fire risk model and associated computations; the WeatherDataHarvesting (WDH) module implementing the fetching of weather data (observations and forecasts) from cloud-services; and finally the DataModel (DM) for representing weather data and fire risk predictions. This section presents the FRC API and DM.

The main purpose of the data model is to represent the weather data on which the fire risk model operates and the fire risks are being computed. The cloud-based services from which weather data can be fetched provide data in many

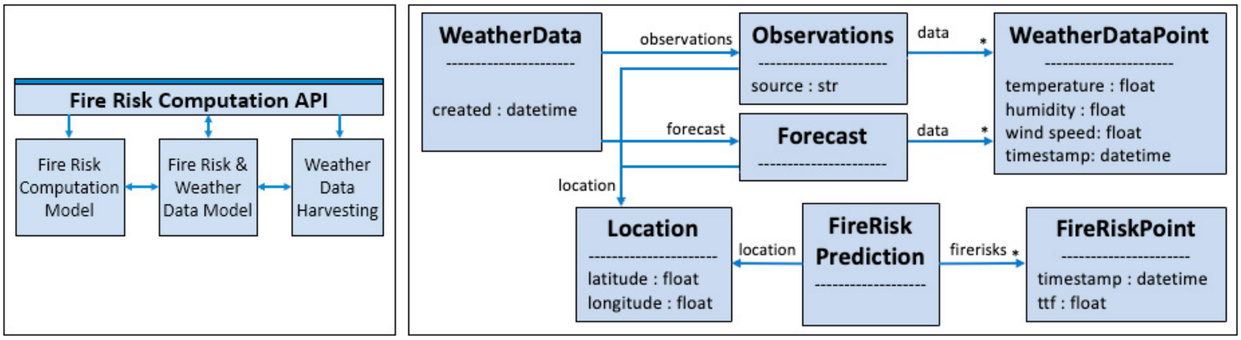


Fig. 1. Overall software architecture of the fire risk computation model (left) and the weather data- and fire risk data model (right).

different formats. Hence, a common internal format is defined, and the WDH module is responsible for converting retrieved weather data into this format by extracting the elements required for the fire risk model. This way, the implementation becomes independent of any cloud-service used to obtain the weather data.

The implementation of the DM is based on the Pydantic library [11], an object-oriented data modelling and validation framework for Python. The data model is shown as a UML class diagram in Fig. 1 (right). The WeatherData class has associated observation and forecast data. The created-attribute is used for traceability to determine when the weather data was obtained. The observations and forecasts are represented by the classes Observations and Forecast, respectively. Both classes contain a location and weather data represented as a list of the WeatherDataPoint objects, i.e., a time series of weather data elements. The Observations class also includes a source-attribute to be able to identify the data origin (via a weather station ID). The WeatherDataPoint class models the parameters required by the fire risk model, herein, temperature, humidity, wind speed (conflagration risk indicator), and a timestamp. The Location class is used to represent geographical locations and is comprised of a latitude and a longitude. The FireRiskPoint class is used to represent the computed fire risks and is comprised of a timestamp and the fire risk indicator, ttf. The FireRiskPrediction class is the main class used to represent the results of a computation, providing a location and a list of the FireRiskPoint objects, i.e., a time series of fire risks.

Listing 1 shows the methods exposed by the FRC API module to applications using the implementation and provided via the `FireRiskAPI` class. The constructor of the class takes a `WeatherDataClient` as a parameter (see Sec. 5), which instantiates the API with the particular methods used to retrieve the weather data. The main method of the API is the `compute_now` method, which, given a location and a specification of the time period for which to retrieve weather data, computes a fire risk prediction. In the simplest use case, all that is required for an application to use the implementation of the fire risk model is to implement a weather data client, create an object of the API class, and then invoke the `compute_now` method with the locations of interest. More details are given in Sec. 5.

Listing 1. Fire Risk Computation API - the main methods exposed to applications embedding the implementation of the fire risk computation API.

```
class FireRiskAPI:
 def __init__(self, client: WeatherDataClient):
        [ ... ]
 def compute_now(self, location: Location,
                    obs_delta: datetime.timedelta) -> FireRiskPrediction:
        [ ... ]
 def compute_now_period(self, location: Location, obs_delta: datetime.timedelta,
                           fct_delta: datetime.timedelta) -> FireRiskPrediction:
        [ ... ]
 def compute_period(self, location: Location, start: datetime,
                       end: datetime) -> FireRiskPrediction:
        [ ... ]
 def compute_period_delta(self, location: Location, start: datetime,
                             delta: datetime.timedelta) -> FireRiskPrediction:
        [ ... ]
```

## 4. Implementation of the Fire Risk Model

The FRC module performs fire risk computations based on observed and forecasted weather data received from the WDH module in accordance with the DM (see Fig. 1 (right)). Received weather data is then validated and interpolated within the FRC module according to the specified timestep needed for the numerical solution procedure. The validation process primarily involves assuring that the weather data is sorted correctly and identifying and quantifying any gaps of missing data so that clients may be warned in case of reduced trustworthiness in the fire risk predictions.

The FRC module implements the fire risk model outlined in Sec. 2 and detailed in [15] with the initial parameters set according to [4]. The computation implemented follows the solution procedure shown in Fig. 2. The first step involves initial conditions and relates to both model- and scenario-specific parameters. Within the present implementation, these quantities are specified in a separate configuration file and must be set according to [4] before using the model for a given geographical region. Following the fire risk computation and prior to constructing a FireRiskPrediction response, the vectors containing the computed fire risks are reduced in size, from progressing according to the fixed computational timestep to instead representing fire risk by the hour.

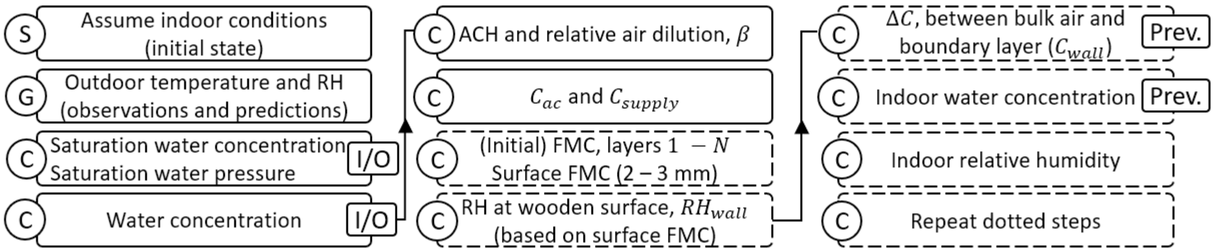


Fig. 2. Solution procedure for modelling in-home RH and in-home wooden FMC. S, G and C refer to whether values at the specific step are Set (assumed), Get (retrieved) or Computed, respectively. I/O refers to computations for both Indoor and Outdoor conditions, while Prev. refers to whether a step depends on values from a previous timestep. The dashed rectangles indicate the iterative steps in the implementation.

## 5. Data Harvesting from Cloud Services

The implementation of the WDH module (see Fig. 1 (left)) is now considered. This module is responsible for retrieving weather data observations and forecasts from cloud-based services and extracting and transforming the required weather data elements into the data model that can be used by the FRC module.

A key property of the present fire risk model implementation is being agnostic to the particular underlying cloud-services or sources from which the weather data elements are obtained. This is partly done via the definition of the data model and partly by providing a specification of interfaces for forecast- and observation clients and for the extraction of weather data elements. By implementing these interfaces, any third-party application can embed and use the implementation of the fire risk model. To provide representative examples, it is shown how these interfaces are implemented for locations in Norway using the REST APIs for forecasts (MET API) [9] and observations (FROST API) [8] provided by the Norwegian Meteorological Institute (MET). Both services provide data in a JSON format.

Listing 2 below shows the WeatherDataClient class specifying the two methods that must be implemented. One method is for fetching observations, and one method is for fetching forecasts. Both methods take a location as a parameter and return an object of class Observations or class Forecast, respectively, as per the data model.

The specific implementations of the fetch-methods are then to be provided in a subclass of WeatherDataClient. Listing 3 below shows an extract from the implementation of the METClient-class, which constitutes a subclass for the weather data services provided by the MET. The complete implementation is available via [13]. The constructor sets up the cloud end-point URL and credentials for accessing the specific MET service used (not shown). The fetch_forecast_raw method then builds and sends the actual HTTP request. The fetch_observations_raw method is similar, but first involves an additional HTTP request to obtain the identity (ID) of the nearest MET weather station that will be the source of the weather data. Then, the station ID becomes the source argument, and the request can be made for a specified time period.

Listing 2. The WeatherDataClient class specifying an interface with the methods fetch_observations and fetch_forecast.

```
class WeatherDataClient:
    @abc.abstractmethod
    def fetch_observations(self, location: Location) -> Observations:
        pass
    @abc.abstractmethod
    def fetch_forecast(self, location: Location) -> Forecast:
        pass
```

Listing 3. The METClient class specifying the fetch_forecast_raw method and the method signature for fetch_observations_raw.

```
class METClient(WeatherDataClient):
 def __init__(self, extractor: Extractor):
    [ ... ]
 def fetch_forecast_raw(self, location: Location):
    parameters = {'lat': str(location.latitude), 'lon': str(location.longitude)}
    response = self.send_met_request(parameters)
    return response
 def fetch_observations_raw(self, source: str,
                            start: datetime.datetime, end: datetime.datetime):
    [ ... ]
```

The weather data returned by the fetch-methods of the client may be of different formats depending on the particular cloud-service being used. The purpose of the extractor interface shown in Listing 4 is to specify the two methods that implementations must provide to make the fetched weather data conform to the data model. The data parameter in both methods will be the response returned by a WeatherDataClient.

Listing 4. The Extractor class specifying the interface for weather data extraction and transformation.

```
class Extractor:
    @abc.abstractmethod
    def extract_observations(self, data: str) -> Observations:
        pass
    @abc.abstractmethod
    def extract_forecast(self, data: str) -> Forecast:
        pass
```

## 6. Validation and Performance Evaluation

This section considers the validation and performance evaluation of the implementation in terms of computed fire risks, as well as the computation and communication costs involved in computing fire risks and retrieving weather data from cloud services. Since an existing fire risk model has been implemented, the implementation needs to be verified and validated, i.e., compared with baseline modelling results (intermittent calculations and indoor RH) and in-situ measurements (indoor RH), respectively. The data computed and collected in [15] was used to verify and validate the implementation. A representative old wooden home built in 1920 in Haugesund, Norway, at N 59.4140 and E 5.2701, was used to indicate a mean absolute percentage error (MAPE) [6]. Comparing results from the implementation against baseline modelling results gave a MAPE of 0.05 % considering indoor RH for the particular home. Indicating the accuracy of the implementation by comparing its computed indoor RH with in-situ measured data from [15] resulted in a MAPE of 6.28 %, which is very good for a predictive model [6].

Further validation involved consideration of several wooden homes in the Haugesund region, where measured indoor RH was plotted against the computed indoor RH with weather data harvested from the MET service, see Fig. 3 (left/top). It can be seen that the computed RH for the model implementation fits well with the measurements inside the different homes, especially for lower-value RH. Considering the percentiles of indoor RH for the home built in 1920, Fig. 3 (right) shows that the implementation has a higher number of days at low indoor RH compared to the indoor measurements during the winter of 2016. Since the modelling approach implemented uses a generic enclosure, it will not fit all the different homes, but it is tuned to represent lower values of indoor RH, hence increased risks.

Computing the risk indicator (TTF) for all the homes in Fig. 3 based on the respective indoor measurements (temperature and RH) results in the bottom left plot. It can be seen that the implementation, using FROST weather data, corresponds well with the computed TTFs for the different homes in the area.

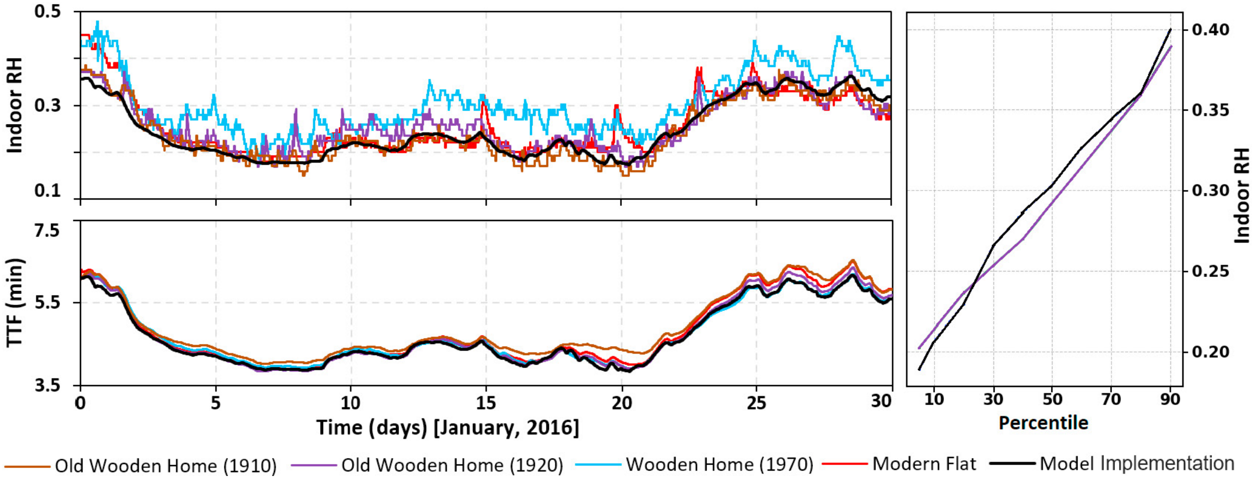


Fig. 3. Plots of indoor RH (measured vs. model implementation) (top/left) and the TTF risk indicator (bottom/left). The percentile plot (right) represents data for the wooden home built in 1910 for December - January (2016).

Computing a fire risk involves the following main steps: (1) fetching weather data observations some specified days into the past (3-5 days); (2) fetching weather data forecast for some specified days into the future; preprocessing the data by extracting and transforming the required data and storing it as per the data model; (3) validate and interpolate the weather data, and finally (4) perform the actual computation of the fire risk.

The cProfile library of Python has been used to evaluate the implementation in terms of computation time. Table 1 provides the collected statistics across ten representative runs of computing fire risks based on weather data three days into the past and nine days into the future. The main steps in the computation, as outlined above, are given in the first column, with the execution times of the individual runs in the subsequent columns. Fetching weather data observations from the FROST service involves two HTTP requests, as described in Sec 5 and is represented by the Station and Observations rows, respectively. The two last columns in the table represent the mean (M) and standard deviation (SD) for the execution times. It can be seen that the Total time spent on obtaining a fire risk is typically around 3-4 seconds, with most time spent on the HTTP request to obtain the Station information. Obtaining the Observation and Forecast weather data are the next two dominating factors, while the Preprocessing of the data and Computation of the fire risk is very efficient. Hence, the dominating factor in computing fire risks is the round-trip communication delays across the Internet required to obtain the weather data. The fire risk computation time itself is very small.

The memory consumption of the fire risk computation is also quite modest. The size of the data retrieved from the weather data cloud services is in the order of Kb's and even further reduced after extracting only the required weather data elements and mapping it into the data model.

## 7. Conclusions

We have presented the implementation and evaluation of a fire risk prediction model for wooden homes exploiting data from cloud-based weather data services. The validation of the implementation ensured that intermittent calculations and computed fire risk predictions complied with the baseline results. Furthermore, computed RH and TTF were found to correspond well with measurements from selected wooden homes in the city of Haugesund. Hence, from the evaluation, it is evident that the cold climate WHFR model has been implemented correctly and can be included in

| Run | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | M | SD |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Station | 2.249 | 2.498 | 2.853 | 2.331 | 2.366 | 2.882 | 2.252 | 2.342 | 2.425 | 2.295 | 2.272 | 0.032 |
| Observations | 0.428 | 0.369 | 1.386 | 0.817 | 0.385 | 0.382 | 0.349 | 0.421 | 0.387 | 0.401 | 0.415 | 0.019 |
| Forecast | 0.285 | 0.227 | 0.277 | 0.231 | 0.234 | 0.237 | 0.234 | 0.243 | 0.221 | 0.228 | 0.257 | 0.040 |
| Preprocess | 0.023 | 0.016 | 0.014 | 0.015 | 0.017 | 0.014 | 0.015 | 0.015 | 0.014 | 0.014 | 0,019 | 0.006 |
| Compute | 0.071 | 0.069 | 0.059 | 0.058 | 0.062 | 0.058 | 0.075 | 0.059 | 0.058 | 0.058 | 0,065 | 0.009 |
| Total | 3.133 | 3.238 | 4.646 | 3.509 | 3.127 | 3.629 | 2.992 | 3.137 | 3.162 | 3.053 | 3.093 | 0.056 |

Table 1. Execution time statistics for fire risk computations. The last two columns present the mean (M) and standard deviation (SD).

other systems to perform fire risk computations for wooden homes. The only precondition for such usage is that the fire risk model parameters are configured according to the context considered.

The evaluation of the implementation shows a very low execution time, within $3-4$ seconds for a single location. It was evident from the evaluation that retrieving weather station ID was the most time-consuming step. Hence, caching of station IDs associated with specific locations may further reduce the execution time by approximately 50 %. In particular, this will reduce the execution time if there is a high number of locations for which fire risk is computed by scheduled services, which is in line with the intended use of the implementation as part of a smart software system for dynamic fire risk predictions. When embedding our implementation into a larger system, the response time for fire risk computations may even be further reduced by caching computed risk indicators for the locations considered. In that case, fire risks only need to be re-computed at predefined time intervals when new weather data becomes available from the underlying weather data cloud services.

**Acknowledgements.** The authors acknowledge the sharing of recorded weather data records (raw data) performed by Torgrim Log in his work [15]. This research was funded by the Research Council of Norway, grant no. 298993, Reducing fire disaster risk through *dynamic risk assessment and management* (DYNAMIC).

## References

[1] Kraaijeveld A., Gunnarshaug A., Schei B., and Log T. Burning rate and time to flashover in wooden 1/4 scale compartments as a function of fuel moisture content. In *8th Int. Fire Science and Eng. Conf., Interflam, London, UK, 4-6 July*, page 553–558, 2016.
[2] Strand R. D., Kristensen L. M., Svendal T., Fisketjøn E. H., and Hussain A. T. A mobile application for wooden house fire risk notifications based on edge computing. In *Proceedings of the 11th World Conference on Information Systems and Technologies, Pisa, Italy, 4-5 April; Springer: London, UK, in press*, Lecture Notes in Networks and Systems, 2023.
[3] Strand R. D., Stokkenes S., Kristensen L.M., and Log T. Fire risk prediction using cloud-based weather data services. *J. Ubiquitous Syst. Pervasive Networks*, 16(1), 2021.
[4] Strand R. D. and Log T. A cold climate wooden home and conflagration danger index: Justification and practicability for norwegian conditions. *Fire*, 6(10), 2023.
[5] Koo E., Pagni P. J., Weise D. R., and Woycheese J. P. Firebrands and spotting in large-scale fires. *Int. J. Wildland Fire*, 19:818–843, 2010.
[6] Juan Montaño, Alfonso Palmer, Albert Sesé, and Berta Cajal. Using the r-mape index as a resistant measure of forecast accuracy. *Psicothema*, 25:500–506, 11 2013.
[7] Netatmo. Netatmo Smart Home API. `https://dev.netatmo.com/apidocumentation`.
[8] Norwegian Meteorological Institute. Frost. `frost.met.no` (accessed 20 June 2023). Historical Weather Data.
[9] Norwegian Meteorological Institute. MET Weather API. `api.met.no` (accessed 20 June 2023). Predicted Weather Data.
[10] Pirsko A. R. and Fons W. L. *Frequency of Urban Building Fires as Related to Daily Weather Conditions*[online]. Technical Report AFSWP-866, US Dep. of Agriculture, 1956.
[11] Colvin S. and contributors. Pydantic. `https://docs.pydantic.dev/latest/`, 2023. Version 2.4.2.
[12] S. Stokkenes, R.D. Strand, L.M. Kristensen, and T. Log. Validation of a predictive fire risk indication model using cloud-based weather data services. *Procedia Computer Science*, 184:186–193, 2021. The 12th International Conference on Ambient Systems, Networks and Technologies (ANT) / The 4th International Conference on Emerging Data and Industry 4.0 (EDI40) / Affiliated Workshops.
[13] R.D. Strand and L.M. Kristensen. Dynamic fire risk model implementation. `https://github.com/selabhvl/dynamic-frcm`, 2023. Version 1.0.
[14] Log T. Cold climate fire risk; a case study of the lærdalsøyri fire, january 2014. *Fire Technol.*, 52(6):1825–1843, 2016.
[15] Log T. Modeling indoor relative humidity and wood moisture content as a proxy for wooden home fire risk. *Sensors*, 19(22), 2019.
[16] Villa V., Paltrinieri N., Khan F., and Cozzani V. Chapter 1 - a short overview of risk analysis background and recent developments. In Paltrinieri N. and Khan F., editors, *Dynamic Risk Analysis in the Chemical and Petroleum Industry*, pages 3–12. Butterworth-Heinemann, 2016.