BRNO UNIVERSITY OF TECHNOLOGY
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FACULTY OF INFORMATION TECHNOLOGY
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

DEPARTMENT OF COMPUTER SYSTEMS
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

ANALYSIS OF DATA TO SOLVE PROBLEMS WITH HUMIDITY IN BUILDINGS

ANALÝZA DAT PRO ŘEŠENÍ PROBLÉMŮ S VLHKOSTÍ V BUDOVÁCH

MASTER'S THESIS
DIPLOMOVÁ PRÁCE

AUTHOR **Bc. KLÁRA NEČASOVÁ**
AUTOR PRÁCE

SUPERVISOR **Doc. Ing. JAN KOŘENEK, Ph.D.**
VEDOUCÍ PRÁCE

BRNO 2019

Vysoké učení technické v Brně
Fakulta informačních technologií

Ústav počítačových systémů (UPSY) Akademický rok 2018/2019

Zadání diplomové práce

Studentka: **Nečasová Klára, Bc.**
Program: Informační technologie Obor: Počítačové a vestavěné systémy
Název: **Analýza dat pro řešení problémů s vlhkostí v budovách**
Analysis of Data to Solve Problems with Humidity in Buildings
Kategorie: Vestavěné systémy
Zadání:

1. Nastudujte problematiku vzniku nadměrné vlhkosti v budovách.
2. Seznamte se s principy fungování systému BeeeOn.
3. Pro vybrané místnosti vyberte vhodnou sadu senzorů pro měření vlhkosti a dalších veličin, které ovlivňují vznik nadměrné vlhkosti.
4. Ve vybraných místnostech nasaďte senzory a proveďte dlouhodobé měření.
5. Navrhněte systém, který bude na základě měřených dat detekovat události spojené se změnou vlhkosti a navrhovat vhodný způsob regulace výměny vzduchu.
6. Navržený systém implementujte a ověřte na reálných datech.
7. V závěru diskutujte dosažené výsledky.

Literatura:

- Dle pokynů vedoucího.

Při obhajobě semestrální části projektu je požadováno:

- Splnění bodů 1 až 3 zadání.

Podrobné závazné pokyny pro vypracování práce viz http://www.fit.vutbr.cz/info/szz/

Vedoucí práce: **Kořenek Jan, Ing., Ph.D.**
Vedoucí ústavu: Sekanina Lukáš, prof. Ing., Ph.D.
Datum zadání: 1. listopadu 2018
Datum odevzdání: 22. května 2019
Datum schválení: 26. října 2018

Abstract

The aim of this work was to solve problems with excessive humidity in buildings using data analysis. The theoretical part of the work deals with impacts of excessive humidity on the health of building occupants and also the condition of the building structure. Data mining methods including classification, prediction, and clustering are described together with model evaluation and selection. The practical part focuses on hardware platform description and measurement scenarios. Key parameters affecting indoor relative humidity are indoor and outdoor temperature and outdoor relative humidity. The long-term measurement of the mentioned parameters was performed using the set of sensors and BeeeOn system. Measured data was used to design a system for event detection related to a humidity change. The approach to air change regulation in the room was based on natural ventilation.

Abstrakt

Cílem práce bylo řešit problémy s nadměrnou vlhkostí v budovách pomocí analýzy dat. Teoretická část práce se zabývá vlivem nadměrné vlhkosti na zdraví obyvatel budov a také na stav budov. Dále jsou popsány metody dolování z dat včetně klasifikace, predikce a shlukování a současně jsou vysvětleny možnosti vyhodnocení a výběru modelů. Praktická část je zaměřena na popis hardwarové platformy a scénářů měření. Klíčovými parametry, které ovlivňují relativní vlhkosti uvnitř budov, jsou teplota v místnosti, venkovní teplota a venkovní relativní vlhkost. Dlouhodobé měření uvedených parametrů bylo provedeno za využití sady senzorů a BeeeOn systému. Naměřená data byla využita pro návrh systému, který detekuje události spojené se změnou vlhkosti. Regulace výměny vzduchu v místnosti byla stanovena na základě přirozené ventilace.

Keywords

Humidity, relative humidity, specific humidity, data analysis, data mining, classification, prediction, BeeeOn sensor.

Klíčová slova

Vlhkost, relativní vlhkost, měrná vlhkost, analýza dat, dolování z dat, klasifikace, predikce, BeeeOn senzor.

Reference

NEČASOVÁ, Klára. *Analysis of Data to Solve Problems with Humidity in Buildings*. Brno, 2019. Master's thesis. Brno University of Technology, Faculty of Information Technology. Supervisor Doc. Ing. Jan Kořenek, Ph.D.

Analysis of Data to Solve Problems with Humidity in Buildings

Declaration

Hereby I declare that this thesis was prepared as an original author's work under the supervision of Mr. Doc. Ing. Jan Kořenek, Ph.D. All the relevant information sources, which were used during preparation of this thesis, are properly cited and included in the list of references.

.....
Klára Nečasová
May 22, 2019

Acknowledgements

I would like to thank my supervisor Doc. Ing. Jan Kořenek, Ph.D. for his patient guidance and valuable advice. This work is part of the research project IoTCloud – Intelligence for IoT systems, which is financed by the Technology Agency of the Czech Republic.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

People spend about 90 % of their life indoors [10]. Therefore it is necessary to maintain healthy indoor environment that is significantly affected by humidity. Interior moisture sources include cooking, showering, and respiration of occupants. Exterior moisture sources, for example, rain-soaked walls and roofs, flooded or damp basement, humid air and ground, can enter a building by air movement, vapour diffusion, capillary suction, and liquid flow [37].

Activity level, clothing, expectations of the occupants, air temperature, radiant temperature, humidity, and airspeed are the key factors that affect thermal comfort. Thermal comfort is defined as a condition of mind which expresses satisfaction with the thermal environment and is assessed by subjective evaluation. It is mainly controlled using a building's heating, ventilation, and air-conditioning systems. The building's architectural design also significantly influences the thermal comfort [20].

No matter what climate people live in, humidity occurs everywhere on Earth and affects the exteriors and interiors of buildings. Humidity is one of the major indicators of a healthy indoor environment. It is often expressed as relative, absolute or specific humidity. Comfortable relative humidity ranges from 30 % to 70 %. It is possible to determine the common symptoms of excessive humidity in buildings. For example, people can suffer from allergy or asthma thanks to airborne dust mites and mould spores that are spread in the excessively humid air. A big difference between indoor and outdoor temperature can cause condensation, when water drops or fog can occur on the window glass. Mould spots and musty odours indicate the presence of mould and mildew that can result in health problems. Increased indoor humidity causes interior damage, e.g. paint blisters, peeling wallpaper or rotting wood [22].

Extremely increased humidity can be eliminated by certain habits during common activities that people do in their everyday life. For instance, taking short, cold showers, drying of clothes outside or cooking on a slow cooker that produces less moisture to the air can contribute to humidity decrease. Next actions reducing excessive humidity include using of a dehumidifier, humidifier turning off, moving plants outside or placing them in one room and replacing a carpet that retains a moisture, bed bugs and dust mites. Using exhaust, ceiling or box fans, ventilation systems or cracked windows encourages proper ventilation that limits increased humidity. The cheap solution is also natural ventilation. Cleanness of gutters and insulation of pipes are the next important factors eliminating excessive humidity. A significant difference between indoor and outdoor temperature can cause an indoor humidity rise. Apart from the humidity, temperature, airspeed inside a room and personal parameters also influence a healthy indoor environment [22].

A humid environment encourages mildew, mould and bacterial growth that negatively influence indoor air quality. Ventilation is able to remove pollutants and humidity forming indoors or reduce their concentrations to admissible levels for the occupant health and comfort. It should be energy efficient, preserve indoor air quality and it should not harm the occupants or the building. Choice of ventilation rates is based on epidemiological research, laboratory and field experiments, odour perception, irritation, occupant preferences, productivity, and experience [22].

Humidity is one of the major parameters that influences the use of outdoor air ventilation. Two types of ventilation exist: natural and mechanical. Compared to mechanical ventilation, natural ventilation is energy efficient, requires little maintenance, has low initial costs, and is environmentally friendly. Natural ventilation should be used wherever and whenever is possible apart from areas where the quality of the air outside the house is worse than indoor air quality. Because of its cheapness and simplicity, only natural ventilation is considered in the work [22].

Buildings in the 21st century use modern materials and technologies to minimize the air infiltration. Building materials can absorb moisture when the relative humidity increases, therefore the ventilation is important for the healthy indoor environment. Ventilation systems require careful design, which involves paying attention to the characteristics of the building, its users and the purpose of use. The closed-loop ventilation can be one of the effective solutions. In the simplest form, it is the control of one output variable based on the measurement of an input variable [22].

Old buildings do not have ventilation systems, therefore, the ventilation is performed manually which is ineffective. It is not possible to control the humidity increase or decrease, air quality, etc. The solution is to use a set of sensors to measure the required quantities and report the state indicating the necessity of window opening due to quantity limit overstepping. Sensors themselves do not offer any information relating to how long a window should be open or if is it appropriate to ventilate in given time [22].

The goal of this thesis is to study problems of excessive humidity in buildings and to acquaint with BeeeOn system. It is necessary to select a suitable set of sensors for performing long-term measurements. Measured data has been evaluated to design a system detecting events related to a humidity change. Then a proper way of air change rate regulation has been defined.

The work is divided into seven chapters. The second chapter is concerned with humidity in buildings. Explanation of classification including selected classification methods, prediction, and clustering are included in the third chapter. The fourth chapter focuses on classifier performance evaluation, model testing, and classifier comparison. The fifth chapter describes the hardware platform and measurement scenarios. The sixth chapter discusses the results of long-term measurement, including a description of the designed system for detection of events related to humidity changes based on the mentioned measurements. An appropriate way for regulation of air change rate was also designed. The last chapter concludes the work and proposes further improvements.

Chapter 2

Humidity in Buildings

Humidity is an important factor affecting a healthy indoor environment. Main reasons for humidity control are health, comfort, and safety and maintenance of building materials. Excessive relative humidity can cause mould, mildew, illnesses, and damage to building materials. Therefore it is essential to maintain humidity within a safe range [22].

Water in the atmosphere occurs as a gas (water vapour from evaporation), as a liquid (small drops of rain), and as a solid (snow and ice) [33]. Humidity can be measured by hygrometers. The commonly used hygrometers are a relative humidity sensor, condensation principle hygrometer, "dew-point" probe, psychrometer (wet-bulb and dry-bulb hygrometer) and mechanical hygrometer. Relative humidity sensor (electrical impedance) is a hygrometer using an electronic element which changes electrical impedance (resistance or capacitance) based on the amount of absorbed water vapour from humid air. It is usually represented as a "probe" fastened either directly or using a cable to an electronics unit to show relative humidity values. Condensation principle hygrometer utilizes cooling to cause controlled condensation that appears in the case of the stable temperature called dew point or frost point. It is realized by a cooled mirror and optical detection of condensation that serves as feedback and enables the mirror temperature control. Calibrated condensation hygrometers characteristically have good resolution and good long-term stability. Therefore they are utilized as references for calibration. "Dew-point" probe is a hygrometer type using an electronic sensor that changes its electrical capacitance according to absorbed water. It is able to measure water vapour trace levels in very dry gases. The hygrometer is called "dew-point" as units of dew point (frost point) are often used to display values. Psychrometer (wet-bulb and dry-bulb hygrometer) is based on evaporative cooling representing a measure of humidity. A dry thermometer and thermometer wrapped up in a wet wick and placed in moving air are compared. The humidity can be found using tables or by calculation according to the temperature values measured by both thermometers. Modern psychrometers are able to internally calculate and display humidity values. Mechanical hygrometer measures humidity by a change in organic material, for example, change in hair length (hair hygrometer). Results can be recorded on a chart driven using batteries or clockwork. Mechanical hygrometers are used for room monitoring [3].

2.1 Calculation Equations for Humidity

The term relative humidity is often used in media and weather forecasts [22]. It can be expressed using vapour partial pressure, vapour density or vapour mass. The first option is to express **relative humidity** R_H [%] as the ratio of vapour partial pressure in the air to the saturation vapour partial pressure that depends on temperature (see Figure 2.1)

$$R_H = \frac{p_w}{p_{ws}} 100\,\%, \tag{2.1}$$

where p_w is the vapour partial pressure [Pa] and p_{ws} is saturation vapour partial pressure at the actual dry bulb temperature[1] [Pa] [29].

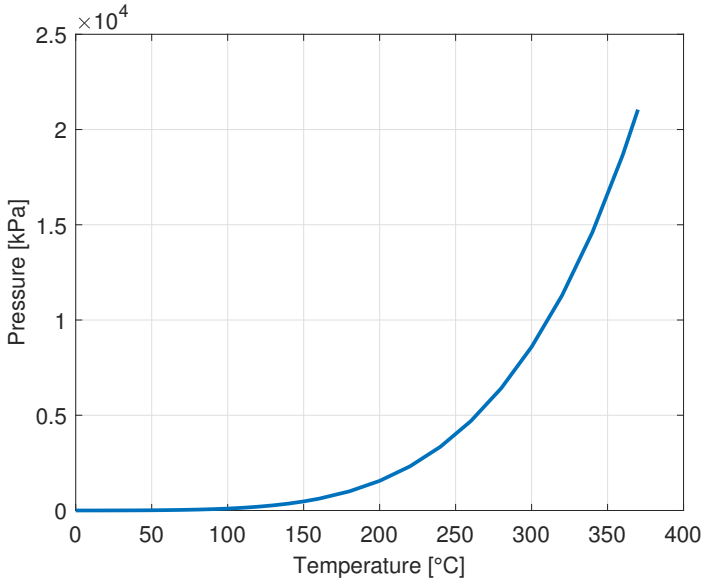

Figure 2.1: Dependency of saturation vapour partial pressure on temperature

Several equations exist to estimate saturation water vapour pressure p_{ws} for a given temperature. The Hyland and Wexler equation [23] defines the saturation water vapour pressure over ice for the temperature T in the range of $-100\,°\mathrm{C}$ to $0\,°\mathrm{C}$ as

$$\ln p_{ws} = \frac{C_1}{T} + C_2 + C_3 T + C_4 T^2 + C_5 T^3 + C_6 T^4 + C_7 \ln T\,, \tag{2.2}$$

where

$$\begin{aligned} C_1 &= -5.6745359\,\text{E+03}\,, \\ C_2 &= 6.3925247\,\text{E+00}\,, \\ C_3 &= -9.6778430\,\text{E-03}\,, \\ C_4 &= 6.2215701\,\text{E-07}\,, \\ C_5 &= 2.0747825\,\text{E-09}\,, \\ C_6 &= -9.4840240\,\text{E-13}\,, \\ C_7 &= 4.1635019\,\text{E+00}\,. \end{aligned}$$

[1] Dry bulb temperature relates basically to the ambient air temperature. The reason why it is called "Dry bulb" is that the air temperature is indicated using a thermometer not influenced by the air moisture [6].

The saturation water vapour pressure p_{ws} over liquid water for the temperature T in the range from 0 °C to 200 °C is defined as

$$\ln p_{ws} = \frac{C_8}{T} + C_9 + C_{10}T + C_{11}T^2 + C_{12}T^3 + C_{13}\ln T\,, \tag{2.3}$$

where

$$\begin{aligned} C_8 &= -5.8002206\,\text{E+03}\,, \\ C_9 &= 1.3914993\,\text{E+00}\,, \\ C_{10} &= -4.8640239\,\text{E-02}\,, \\ C_{11} &= 4.1764768\,\text{E-05}\,, \\ C_{12} &= -1.4452093\,\text{E-08}\,, \\ C_{13} &= 6.5459673\,\text{E+00}\,. \end{aligned}$$

For the temperature T in the range of 0 °C to 80 °C with the error less than 0.1 %, the water vapour saturation pressure p_{ws} is defined as [43]

$$\ln p_{ws} = 23.58 - \frac{4044.2}{235.6 + T}\,. \tag{2.4}$$

Next possibility is to use the Antoine equation that is one of the least complex in comparison with the Hyland and Wexler equation, Geoff-Gratch equation, the Arden Buck equation, the Sonntag formula and the Magnus formula that are more complicated but provide better accuracy. The calculation of the saturation water vapour pressure using the mentioned equations can be inaccurate due to rounding errors. The most accurate method to find out the water vapour partial pressure at a given temperature is to use the look-up table that was experimentally determined [4, 40, 23].

Another equation for expressing relative humidity R_H is defined as the ratio of the vapour density of the air to the saturation vapour density

$$R_H = \frac{\rho_w}{\rho_{ws}} 100\,\%\,, \tag{2.5}$$

where ρ_w is vapour density [kg·m^{-3}] and ρ_{ws} is vapour density at saturation at actual dry bulb temperature [kg·m^{-3}] [22].

Relative humidity R_H can be also calculated as the ratio of an actual mass of water vapour in a given air volume to the mass of water vapour required to saturate at this volume

$$R_H = \frac{m_w}{m_{ws}} 100\,\%\,, \tag{2.6}$$

where m_w is mass of water vapour in the given air volume [kg] and m_{ws} is mass of water vapour required to saturate at the volume [kg] [22].

As water vapour concentration grows, the rate of condensation increases. The amount of water grows until the rate of condensation and the rate of evaporation are in equilibrium.

Then the water vapour concentration will stop increasing because the air is saturated with water vapour. Relative humidity of saturated air is 100 %. The temperature to which the air has to be cooled to reach saturation is called dew point $T_{dewpoint}$ [°C] and it is defined as

$$T_{dewpoint} = \frac{b\left(\frac{aT}{b+T}\ln R_H\right)}{a - \left(\frac{aT}{b+T}\ln R_H\right)}, \tag{2.7}$$

where a is a constant and is set to 17.27, b is also a constant and its value is 237.7, R_H is relative humidity in the range from 0 to 1 and T is temperature [°C]. Air is not able to hold the moisture in a gas form at that temperature. It results in the forming of liquid water or dew [32, 22, 28].

Except for relative humidity, humidity can be expressed by absolute or specific humidity. **Absolute humidity** A_H [g·m^{-3}] does not fluctuate with the temperature of the air and it can be calculated as

$$A_H = \frac{m}{V}, \tag{2.8}$$

where m is the mass of the water vapour [g] and V is the volume of the air and water vapour mixture [m^3]. The absolute humidity A_H is also defined using the equation

$$A_H = \frac{6.112\, e^{\left[\frac{17.67T}{T+243.5}\right]} R_H\, 2.1674}{273.15 + T}, \tag{2.9}$$

where T is temperature [°C], R_H is relative humidity [%] and e is the Euler's number ($e \doteq 2.71828$). Over the temperature range from −30 °C to 35 °C, the equation is accurate to within 0.1 % [11]. Warm air can hold more water than cool air. But if the air (warm or cool) is holding half as much moisture as it can hold when saturated, the relative humidity is 50 % [22].

Specific humidity S_H [g·kg^{-1}] is the ratio of the mass of water vapour in a parcel of air to the total mass of the moist air

$$S_H = \frac{\text{mass of water vapour}}{\text{total mass of moist air}} = \frac{m_w}{m_a} = 0.622\frac{p_w}{p - p_w}, \tag{2.10}$$

where p is the atmospheric pressure and p_w is the partial pressure of water vapour. The atmospheric pressure can be calculated as the sum of the partial pressure of dry air p_a and water vapour $p = p_a + p_w$ [Pa] [2]. The previous equation can be rewritten to a formula

$$S_H = 0.622\,\frac{\frac{R_H\, p_{ws}}{100}}{p - \frac{R_H\, p_{ws}}{100}}, \tag{2.11}$$

where R_H is relative humidity [%] and p_{ws} is saturation vapour partial pressure at the actual dry bulb temperature [Pa].

Using equation (2.11), relative humidity can be expressed as

$$R_H = \frac{100\, S_H\, p}{(0.622 + S_H) p_{ws}}, \tag{2.12}$$

where S_H is specific humidity, p is the atmospheric pressure, and p_{ws} is the saturation vapour partial pressure at the actual dry bulb temperature. Specific humidity is approximately equal to the mixing ratio. The mixing ratio M_R [g·kg^{-1}] is defined as the ratio of the mass of water vapour to the mass of dry air [22]

$$M_R = \frac{\text{mass of water vapour}}{\text{mass of dry air}} = \frac{m_w}{m_d}\,. \tag{2.13}$$

The only difference between mixing ratio and specific humidity is that mixing ratio is related to the mass of dry air whereas specific humidity is referred to the moist air (the mixture of the dry air and water vapour). However, both specific humidity and mixing ratio do not take into consideration changes in temperature [9].

2.2 Indoor Humidity Sources and Thermal Comfort

The relative humidity is affected by ventilation and moisture gains that include the outside temperature and humidity. The outdoor temperature is a major aspect of weather observation. It is influenced by sunshine intensity, perception occurrence, and wind speed. The outside humidity, usually referred to as relative humidity, is the other important factor affecting outdoor temperature. The lower the outdoor temperature is, the higher the outside relative humidity tends to be. The heating, cooling and everyday activities including cooking and showering also influence relative humidity as well as room occupants that create quite a lot of moisture in the room, since they exhale the air with a relative humidity of 100 % [22].

Airspeed and thermal radiation are mostly outdoor effects and it is difficult to measure and control them. As a consequence, thermal comfort takes into consideration only temperature and humidity. Comfort zone can be determined using three approaches: comfort zone according to ISO, ASHRAE, and the Heat Index concept [22].

The thermal comfort can be analytically determined and interpreted using a calculation of the PMV (Predicted Mean Vote) and PPD (Predicted Percentage Dissatisfied) indices and local thermal comfort. The human thermal sensation is associated with the thermal balance of a whole body that is affected by personal parameters – physical activity, clothing, and environment parameters – air temperature, mean radiant temperature, relative air velocity, and relative humidity. The factors mentioned above can be estimated or measured. By calculation of the PMV, the thermal sensation for the whole body can be predicted. The information on thermal discomfort or thermal dissatisfaction is provided by the PPD index predicting the percentage of occupants that will be dissatisfied with the thermal conditions in a given environment. The PPD index can be calculated from the PMV. The ISO 7730 and ASHRAE 55 standards are based on PMV method, which is still used in practice [22].

The standard ISO 7730 specifies the ergonomics of the thermal environment [30]. It does not consider that higher temperatures can be acceptable at low humidity. As a result, temperature limits are vertical (see Figure 2.2). Even though temperature ranges are different in each season, the relative humidity is set between 70 % RH in the summer time and 30 % RH in the winter time. These limits are defined to reduce health problems or microbial growth. The described approach is suitable for simpler execution of air-conditioning algorithms used in less complex applications [5].

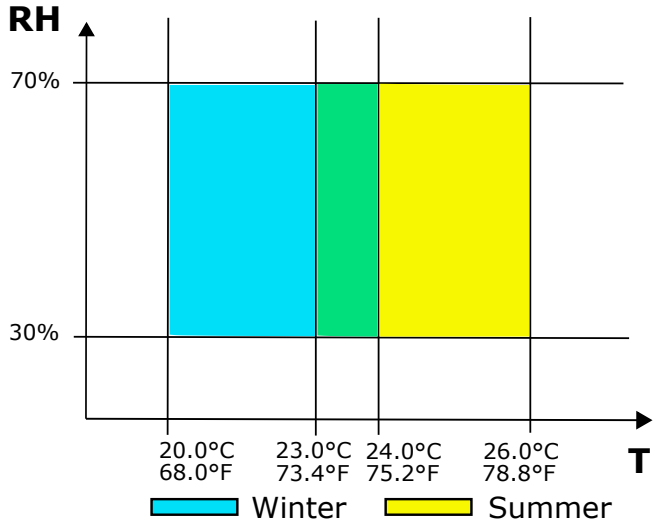

Figure 2.2: RH/T diagram expressing the comfort zone according to ISO 7730

The American Society of Heating, Refrigerating and Air-Conditioning Engineers (ASHRAE's) is the standard that has a method for determining the acceptable thermal condition in occupant-controlled naturally conditioned spaces [34]. It expresses the effect of experiencing higher temperature together with high relative humidity by slanting the boundaries of the temperature limits (see Figure 2.3). Mentioned attitude influences heating, ventilation and air conditioning control systems. Temperature and humidity monitoring in indoor spaces contribute to energy saving because, in the case of low humidity, a higher temperature is admissible. As a result, cooling is not demanded [5].

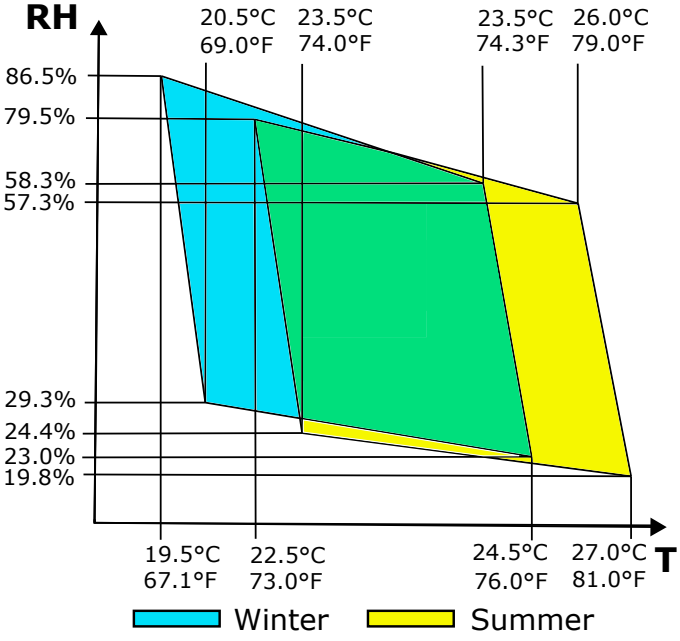

Figure 2.3: RH/T diagram showing the comfort zone according to ASHRAE 55-1992

The heat index (HI) was originally designed for outdoor working conditions, in practice, it is used in indoor and outdoor work environments. The heat index applies to ambient

temperatures of 27 °C or higher and relative humidity of 40 % or higher. If the relative humidity is greater than 85 % and the ambient temperature is between 26.6 °C (80 °F) and 30.5 °C (87 °F), the calculation of the heat index has to be adjusted [36]. The heat index, also known as the apparent temperature, is based on subjective measurements. It expresses what the temperature feels like to the human body when relative humidity is combined with the air temperature. When the humidity level increases, a human will feel warmer. Figure 2.4 shows health impacts for different heat index values [41, 5].

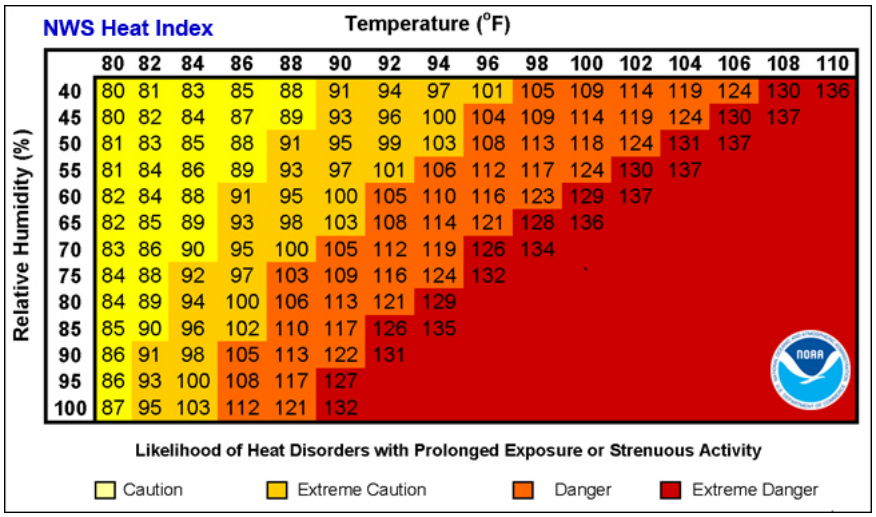

Figure 2.4: Heat index table from National Weather Service (NWS)

As mentioned in chapter 1, the thermal comfort is influenced by relative humidity, temperature, airspeed in the room, outdoor effects and personal parameters. Determining a comfortable relative humidity range is influenced by home location, home construction, the season of the year and occupant sensitivity. Too high humidity (more than 70 %) can contribute to the mould, fungus and dust mites growth that are dangerous for people suffering from allergies and asthma. On the other hand, too low humidity (less than 30 %) causes health problems, for example, dry eyes and upper respiratory tract diseases [22].

Personal parameters such as physical activity, clothing, age, gender, and health affect human comfort but pleasant temperature depending on the relative humidity level should be considered too. The temperature that is comfortable at low relative humidity can be uncomfortable at high relative humidity due to the dropping of evaporating cooling efficiency. As a consequence, humidity control helps to optimize occupants' comfort. External factors like outside temperature and amount of sunshine have also an impact on the temperature in a building [22].

Common activities that people do daily belong to the main causes of high humidity in a flat or house. Taking a shower is one of the major causes. During showering a lot of moisture in the bathroom is created. When moisture mixes with the air, water vapour occurs and comes into contact with cold surfaces (mirror, window, tiles etc.). It results in condensation that becomes evident as small water drops or misted mirror or window. The bigger problem is mould growth that can damage walls. The next cause is cooking, boiling water on the stove or boiling a kettle causing water evaporation and increased humidity in the kitchen. Arisen water vapour can condensate on colder objects. Leaving

aside unpleasant hot conditions in the kitchen, high humidity is not suitable for kitchen walls or furniture [22].

Drying laundry inside the home also falls into major sources of humidity. Wet clothes from one washing cycle can produce up to two litres of water dispersed in the air. Water vapour changes into condensate and mould on the wall appears. Gas heaters add some combustion and water vapour to the air. As a result, high humidity and wet surfaces occur and support dust mites and mould growth that can cause health problems. Plants are the next huge source of moisture that affects humidity level in a flat or house. Finally, insufficient ventilation can contribute to mould growth because apart from above-mentioned activities, occupants also encourage humidity rise by exhaling water vapour [22].

Chapter 3

Data Mining Methods

Data mining is a tool for data analysis which replaces tools based on statistics due to their inability to discover the required knowledge in such large volumes of data. Data mining is defined as the extraction of interesting (non-trivial, hidden, formerly unknown and potentially useful) data models and patterns from large data volumes. Mentioned models and patterns represent knowledge obtained from the data [42]. Data analysis is a process of inspecting, cleaning, transforming, and modelling data with the goal of discovering useful information, suggesting conclusions, and supporting decision making. It concentrates on knowledge discovery for predictive and descriptive purposes to reveal new ideas or to confirm existing ideas [17]. The classification, prediction and clustering are data mining methods. Section 3.2 and 3.3 are based on [8, 42].

3.1 Feature Selection

In real-world classification tasks, many problems can occur, for example, irrelevant and redundant features, the high-dimensional data or high dependency of features [16]. An irrelevant feature influences the learning process even though it is not directly related to the target concept. A redundant feature adds nothing to the target concept. The elimination of irrelevant or redundant features leads to a decrease in running time and a more general classifier. The high-dimensional data results in overfitting of learning model and performance degradation. One of the dimensionality reduction techniques is feature selection. Its aim is to select an appropriate small subset of the features from the original data set according to a particular relevance evaluation criterion. The evaluation criterion contributes to the better learning performance, lower computational cost and better model interpretability. The feature selection method includes four steps: subset generation, subset evaluation, stopping criterion, and result validation as shown in Figure 3.1. During the subset generation, the candidate feature subset is selected on the basis of a certain search strategy, then it is evaluated using a given evaluation criterion in the second step. After the stopping criterion is met, the subset of all candidates that best fits the evaluation criterion is selected [35]. Frequently used stopping criteria are [19]:

- the search completes,
- a specified bound is reached (maximum number of iterations or minimum number of features),
- additional (deletion) of any feature does not result in a better subset,

- a selected subset meets the requirements (e.g., the classification error rate is less than the required error rate).

Finally, the result validation performs validation of the chosen subset using the validation set [35].

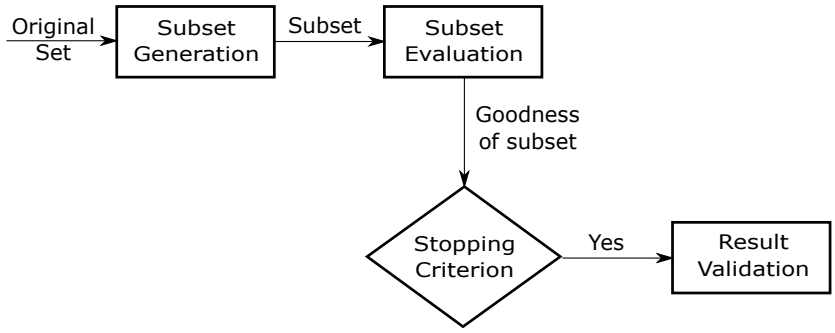

Figure 3.1: Scheme of feature selection

A general feature selection for classification is shown in Figure 3.2. The training classification phase is significantly influenced by feature selection. After the feature generation is performed, a subset of features is chosen and then passed as an input to the learning algorithm. The feature selection can be independent of the learning algorithm (filter models), or it can iteratively use the learning algorithm performance to determine the quality of the chosen features (wrapper models). A classifier using selected features is induced for the prediction phase. Feature selection for classification typically focuses on a selection of the minimal feature subset on the basis of two criteria [35]:

- the accuracy of classification does not greatly decrease,
- the class distribution based on the selected feature values is similar to the original class distribution based on all features.

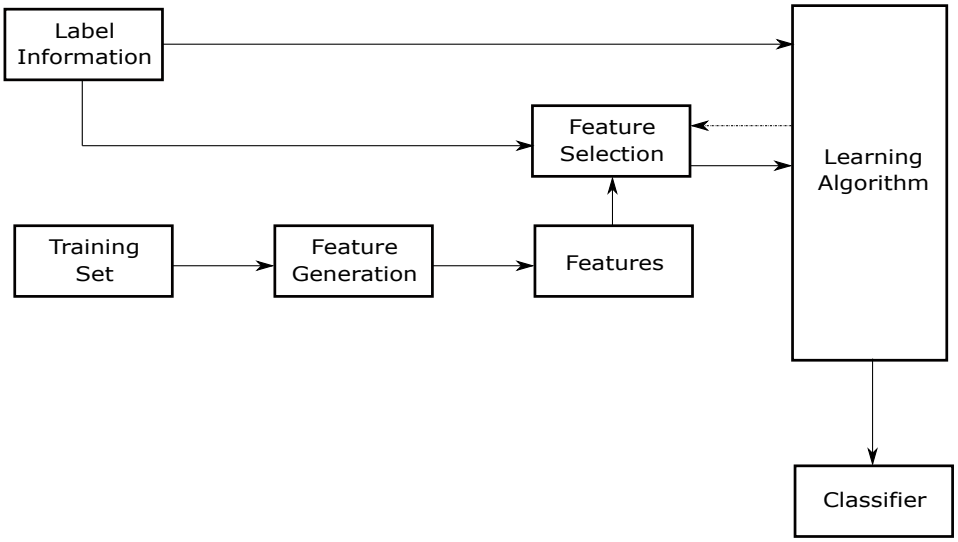

Figure 3.2: Scheme of feature selection for classification

Discovering of the optimal feature subset by the exhaustive search can be significantly expensive because 2^n possible subsets can be created using n features. The aim of heuristic or random search methods is to decrease computational complexity by compromising performance. Both methods use a stopping criterion to avoid an exhaustive search of subsets [35].

3.1.1 Filter models

Filter model evaluates features using only the data characteristics without involving any classification algorithms. It implies feature selection is performed only once and subsequently various classifiers can be evaluated. The model includes two steps (see Figure 3.3). The first step selects features on the basis of measures, e.g., information, distance, dependence, consistency. During the second step, the training data is used to learn a classifier, then the classifier is tested on the testing data. One of the several filter model characteristics is that it relies on intrinsic data properties. As a consequence, different classifiers can be learned using the chosen features. Unlike measuring classifier accuracy, measuring information gain, distance, dependence, or consistency is typically less expensive (in time complexity) which results in a faster subset production. The filter model can also deal with larger data set than classifier because of the measure simplicity and low time complexity. It is possible to use a filter model to reduce data dimensionality so that the classifier can be learned using the low-dimensional data. The main disadvantage is that these filter methods do not interact with the classifier and most methods are univariate. Univariate methods ignore feature dependencies which can result in worse classification performance in comparison with other feature selection methods. On the other hand, multivariate methods solve the problem by involving feature dependencies of a certain degree [31, 18, 35].

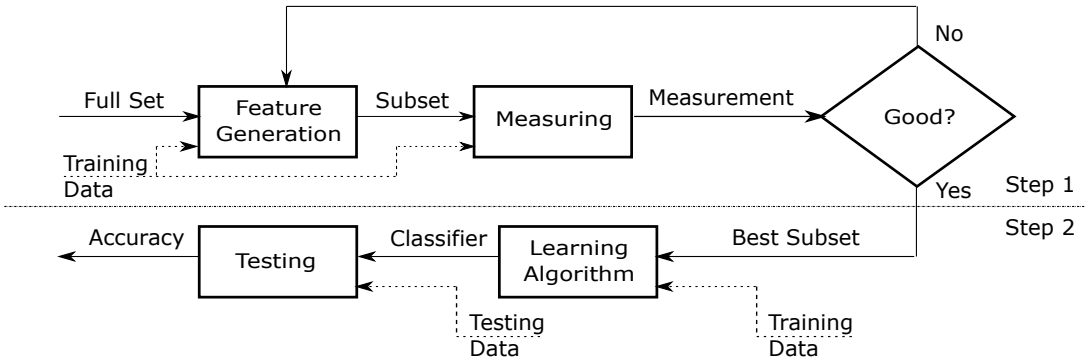

Figure 3.3: Scheme of filter-based feature selection

The basic filter-based feature selection metrics are information gain [42, 8], gain ratio [42, 8], Pearson correlation [8], mutual information [24], χ^2 (chi-square) test [8], Fisher score [1] or Euclidean distance [8].

Let $A = \{a_1, a_2, \ldots, a_m\}$ is a set of attribute numeric values of A with probabilities of occurrence $p_1, p_2, \ldots, p_m$. The average value of the given attribute can be calculated using equation

$$\overline{A} = \sum_{i=1}^{m} p_i a_i. \tag{3.1}$$

The entropy expresses an average information content of a given attribute defined by equation (3.2). The maximal entropy is reached if all attributes appear with the same probability. If an attribute a_i occurs with probability $p_i = 1$, the entropy is minimal

$$H(A) = \sum_{i=1}^{m} p_i H(a_i) = -\sum_{i=1}^{m} p_i \log_2(p_i). \tag{3.2}$$

Let S is a set of all samples (training set), C_i represents individual classes ($i = 1, \ldots, m$) and s_i denotes the number of samples of S that are classified to the class C_i ($i = 1, \ldots, m$). The expected information needed to classify a sample in training set S is

$$H(S) = -\sum_{i=1}^{m} p_i \log_2(p_i)\,, \tag{3.3}$$

where $p_i = \frac{s_i}{|S|}$. Let a_j represents individual values of a given attribute A ($j = 1, \ldots, v$), S_j denotes the set containing only samples from S, whose attribute A has a value a_j ($j = 1, \ldots, v$) and s_{ij} is a number of samples of S_j classified in class C_i ($i = 1, \ldots, m$; $j = 1, \ldots, v$). The expected information necessary to classify a sample from S based on partitioning according to A is defined as

$$H_A(S) = \sum_{j=1}^{v} \frac{|S_j|}{|S|} H(S_j)\,. \tag{3.4}$$

The difference between the original information requirement and the new requirement is called information gain

$$\mathit{Gain}(A) = H(S) - H_A(S)\,. \tag{3.5}$$

If the attribute A is unique, then $H_A(S) = 0$ and that partitioning is needless for classification. It is possible to normalize information gain using a "split information" value which can be calculated similarly as $H(S)$

$$\mathit{SplitH}_A(S) = -\sum_{j=1}^{v} \frac{|S_j|}{|S|} \log_2\left(\frac{|S_j|}{|S|}\right)\,. \tag{3.6}$$

The training data set S is divided into v parts which correspond to the v outcomes of a test on attribute A. This splitting of S generates the potential information represented by $\mathit{SplitH}_A(S)$. The gain ratio can be expressed as

$$\mathit{GainRatio}(A) = \frac{\mathit{Gain}(A)}{\mathit{SplitH}_A(S)}. \tag{3.7}$$

The correlation between two numeric attributes can be expressed using the Pearson correlation coefficient

$$r_{A,B} = \frac{\sum_{i=1}^{n}(a_i - \overline{A})(b_i - \overline{B})}{n\sigma_A\sigma_B} = \frac{\sum_{i=1}^{n}(a_i b_i) - n\overline{A}\,\overline{B}}{n\sigma_A\sigma_B}\,, \tag{3.8}$$

where A and B are numeric attributes, a_i is the value of A in tuple i, b_i is the value of B in tuple i, n is the number of tuples, $\overline{A}$ is the mean value of A, $\overline{B}$ is the mean value of B, σ_A is standard deviation of A, σ_B is standard deviation of B and $\sum(a_i b_i)$ is the sum of product

of the value of A in tuple i and the value of B in tuple i. Value of Pearson correlation coefficient ranges from -1 to 1. Attributes A and B are positively correlated, if $r_{A,B}$ is greater than 0. Positively correlated means that the values of A increase as the values of B increase. The higher the coefficient is, the stronger the correlation is. If attributes are strongly correlated, then they are redundant and they can be eliminated. The attributes A and B are independent if $r_{A,B}$ is equal to 0. The attributes A and B are negatively correlated, if $r_{A,B}$ is less than 0. Negatively correlated means that the values of A increase as the values of B decrease. The correlation between attributes can be also expressed by scatter plots. If A and B are correlated, then A does not have to cause B or B does not have to cause A.

Mutual information uses entropy defined by the previously mentioned equation (3.2) to express the average information content of an attribute. If an attribute X is observed, then the conditional entropy can be calculated as

$$H(Y/X) = -\sum_{x}\sum_{y} P(x,y)\log P(y/x)\,. \tag{3.9}$$

In the case of observation of the attribute X, the uncertainty in the attribute Y is decreased which is defined by

$$I(Y,X) = H(Y) - H(Y|X)\,. \tag{3.10}$$

The value of $I(X,Y)$ corresponds with mutual information between Y and X. If mutual information is equal to 0, X and Y are independent. If mutual information is greater than 0, they are dependent which means that one attribute can provide information about the other. The mentioned equations are valid for discrete attributes but the same formulas can be defined for continuous attributes supposing that the summations are replaced with integrations.

The correlation between two nominal attributes, A and B, can be expressed using χ^2 test. Assume that A has c different values $a_1, a_2, \ldots, a_c$ and B has r different values $b_1, b_2, \ldots, b_r$. A together with B defines data tuple that can be represented as a contingency table. The table rows are created by the r values of B and the table columns are defined by the c values of A. The table cell represents the joint event marked as (A_i, B_j), where a_i is the value of A and b_j is the value of B. The χ^2 value is calculated as

$$\chi^2 = \sum_{i=1}^{c}\sum_{j=1}^{r}\frac{(o_{ij}-e_{ij})^2}{e_{ij}}\,, \tag{3.11}$$

where o_{ij} is the observed frequency (actual count) of the (A_i, B_j), and e_{ij} is the expected frequency of (A_i, B_j) that can be expressed as

$$e_{ij} = \frac{count(A = a_i) \times count(B = b_j)}{n}\,, \tag{3.12}$$

where $count(A = a_i)$ is the number of tuples where the attribute A is set to a_i, $count(B = b_j)$ is the number of tuples where the attribute B is set to b_j, and n is the number of data tuples. If the actual count significantly differs from the expected one, then the corresponding table cell contributes to the χ^2 value the most. The χ^2 statistic tests the hypothesis that no correlation occurs between attribute A and attribute B which means that they are independent. The test uses a significance level considering $(r-1)\times(c-1)$

degrees of freedom. The attribute A and attribute B are statistically correlated if the hypothesis can be rejected.

The Fisher score is naturally proposed for numeric attributes and it expresses the ratio of the average interclass separation to the average intraclass separation using the equation

$$F = \frac{\sum_{j=1}^{k} p_j(\mu_j - \mu)^2}{\sum_{j=1}^{k} p_j \sigma_j^2}, \tag{3.13}$$

where μ_j is the mean of data points of class j for a certain feature, σ_j is the standard deviation of data points belonging to class j for a certain feature, p_j is the number of data points of class j, μ is the global mean of the data on the evaluated feature. The larger the value of the Fisher score is, the more appropriate the attribute for the classification algorithm is.

The Euclidean distance is another commonly used metric. Let $X_1 = (x_{11}, x_{12}, \ldots, x_{1n})$ and $X_2 = (x_{21}, x_{22}, \ldots, x_{2n})$ be two attributes, then the distance is

$$dist(X_1, X_2) = \sqrt{\sum_{i=1}^{n} (x_{1i} - x_{2i})^2}. \tag{3.14}$$

3.1.2 Wrapper models

In comparison with filter models, wrapper models use a certain classifier to evaluate the quality of chosen features and to solve the problem of feature selection without considering of the selected learning machine. The wrapper model includes two steps (see Figure 3.4).

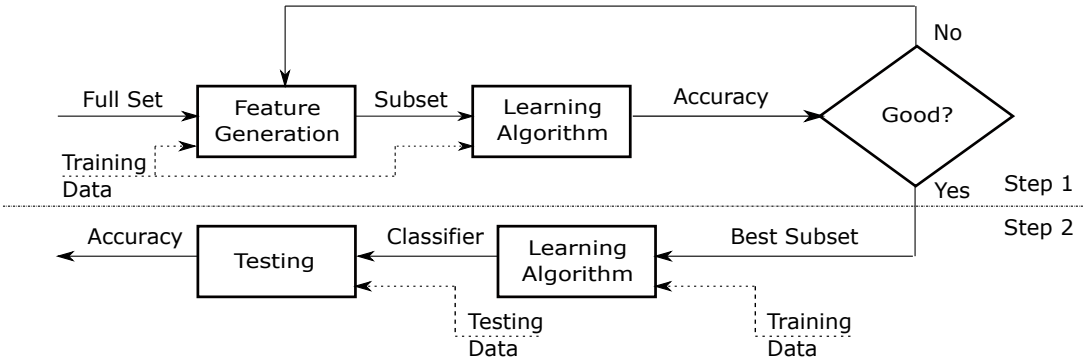

Figure 3.4: Scheme of wrapper-based feature selection

The first step ensures the systematic generation of feature subsets on the basis of the selected search direction. For each feature subset, a classifier is generated from the data containing the selected features. Accuracy of the classifier is stored and only the feature subset with the highest accuracy is maintained. At the end of the first step, the subset with the highest accuracy is selected. Within the second step, the training data with the best feature subset is used to learn a classifier, then the testing data is utilized for testing the classifier and the predictive accuracy evaluation. Consideration of feature dependencies and the interaction between feature subset search and model selection are advantages of wrapper models. On the contrary, wrapper models are more predisposed to overfitting than filter models. The biggest challenge is to truly estimate the accuracy of a classifier because its accuracy on the training data can differ from the accuracy on the testing data. It is possible to use

cross-validation but then the feature selection is more time-consuming. The next challenge is a classifier selection. The classifiers that are time demanding should not be considered as during the first step, a classifier is created for each generated feature subset. It is possible to distinguish two classes of heuristic search methods that try to find an optimal subset: deterministic and randomized search algorithms [31, 18, 35].

Wrapper methods can be divided into two groups – sequential selection algorithms and heuristic search algorithms. Common sequential selection algorithms include sequential feature selection, sequential backward selection and their combination. Heuristic search algorithms are for example simulated annealing, randomized hill climbing or genetic algorithms [39, 7, 15, 31].

At the beginning of the sequential feature selection (SFS) algorithm, a feature subset is empty. Then a feature that reaches the highest value for the objective function like classification accuracy is added to the subset and it is removed from the original feature set. Next features are selected in the same way from the remaining features. The algorithm stops if a required number of features is obtained. The dependency between the features is not considered, therefore the algorithm is called a naive SFS algorithm [24].

Unlike the sequential feature selection, a sequential backward selection (SBS) algorithm begins with a set of all features and it eliminates an irrelevant feature. It means that a feature whose elimination causes the lowest decrease in predictor performance is removed [24].

The combination of SFS and SBS is the sequential floating forward selection (SFFS) algorithm which is more flexible than the naive SFS because of the introduction of a backtracking step. In the first step, the algorithm adds a feature to a feature subset which corresponds with SFS and in the second step it excludes a feature from a subset which is in accordance with SBS. Subsequently, it evaluates the newly obtained subset. If the objective function value increased after the feature excluding, the feature is eliminated from the subset and algorithm continues doing the first step with the smaller subset. Otherwise, the algorithm is performed from the top. The described process is terminated if a subset includes required number of features or a required performance is reached [24].

3.1.3 Embedded Models

Filter and wrapper models are similar in the first step, and they do not differ at all in the second step. The model includes four parts: feature generation, feature evaluation, stopping criteria, and testing (see Figure 3.5).

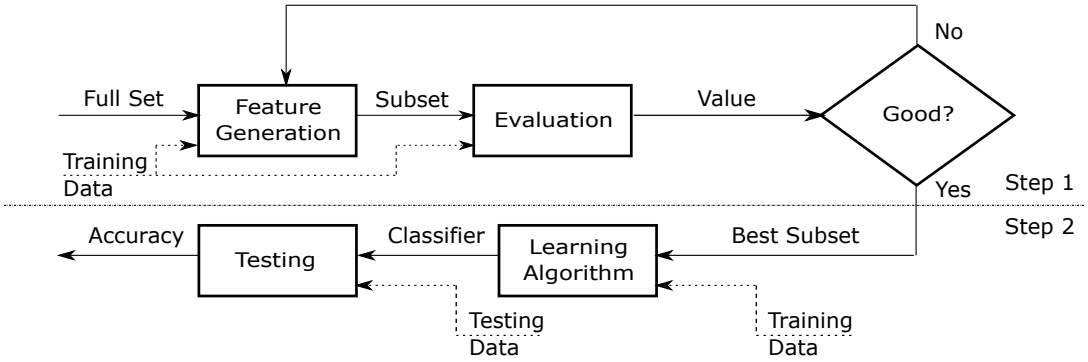

Figure 3.5: Scheme of embedded-based feature selection

Embedded model embeds feature selection with classifier construction. It uses the advantage of wrapper models – it uses the classification model and filter models – it is considerably less time consuming than wrapper models. Embedded models can be divided into three categories. The first category includes pruning methods that consider all features to model training at first, then they set some coefficients to 0 to remove certain features while preserving model performance. The second one contains models with built-in mechanism for feature selection, e.g., ID3 and C4.5 used by decision trees (see subsection 3.2.1). The third category is regularization models using objective functions to decrease fitting errors as much as possible and they also attempt to minimize coefficient values. Features with coefficients close to 0 are removed [35, 18].

3.2 Classification

Classification is generally a process that allows to assign data based on their properties into finite number of classes. It can be also defined as classifying a given object into a certain class based on its properties. By contrast, a prediction is a process that in general assigns values to the continuous character data. In other words, it is predicting a certain value (generally continuous) for the object based on its properties. The classification includes three phases: training, testing, and usage. The goal of the first phase is to determine classification "rules" (to create a classification model) on the basis of training patterns (class membership is known). During the second phase, the created model is tested on other data patterns to evaluate accuracy. After the last phase, the model can be used for unknown data classification.

Various data adjustments can help to make classification and prediction more effective. The basic adjustments include data cleaning, data reduction and data transformation including data normalization. Data cleaning removes noisy data and missing values. Data reduction reduces the data size using, for example, aggregating, eliminating redundant features, or clustering. Data transformation converts data to a suitable format for data classification. Typically, data transformation is used to divide data into certain intervals. Normalization is a strategy of data transformation, where the attribute values are scaled to fall within a smaller range, for example, $\langle 0, 1\rangle$ or $\langle -1, 1\rangle$.

The well-known classification methods are decision tree, naive Bayes classifier, Bayesian network, neural network, Support Vector Machines (SVM) and K-Nearest Neighbours (KNN).

3.2.1 Decision Tree

The decision tree is a directed acyclic graph where each internal node represents the test of a certain attribute value, and the leaf nodes represent the class label. The decision tree requires only discrete attributes and it can be created using the following algorithm.

Algorithm 1 Generate a decision tree from the training tuples of data partition [8].

Input:

- D – data partition, which is a set of training tuples and their associated class labels,
- *attribute_list* – the set of candidate attributes,
- *Attribute_selection_method* – a procedure to determine the splitting criterion that "best" partitions the data tuples into individual classes. The criterion consists of a *splitting_attribute* and, possible, either a *split-point* or *splitting subset*.

Output: A decision tree.

Method:

1: create a node N;
2: **if** tuples in D are all of the same class, C, **then**
3: return N as a leaf node labeled with the class C;
4: **end if**
5: **if** *attribute_list* is empty **then**
6: return N as a leaf node labeled with the majority class in D;
7: **end if**
8: apply ***Attribute_selection_method(D, attribute_list)*** to find the "best" *splitting_criterion*;
9: label node N with *splitting_criterion*;
10: **if** *splitting_attribute* is discrete-valued **and** multiway splits allowed **then**
11: *attribute_list* ← *attribute_list* - *splitting_attribute*;
12: **end if**
13: **for each** outcome j of *splitting_criterion* **do**
14: let D_j be the set of data tuples in D satisfying outcome j;
15: **if** D_j is empty **then**
16: attach a leaf labeled with the majority class in D to node N;
17: **else**
18: attach the node returned by ***Generate_decision_tree(D_j, attribute_list)*** to node N;
19: **end if**
20: **end for**
21: return N;

The *Attribute_selection_method* determines the splitting criterion that selects an attribute (so-called splitting attribute) that "best" separates samples into the individual classes. The attribute can be chosen using several algorithms. One of them is the ID3 (Iterative Dichotomiser 3) algorithm that uses entropy and information gain (see section 3.1.1).

The choice of splitting attribute at node N is the attribute A with the highest information gain $Gain(A)$ (see Equation (3.5)). Equivalently, it is possible to choose the attribute A with the lowest value of the $H_A(S)$ (see Equation (3.4)).

The information gain is influenced by a number of values of attribute A. If the attribute A is unique, it holds that $H_A(S) = 0$. Partitioning is then useless for classification because that attribute could be preferred to the other attributes. The described problem solves C4.5 algorithm that is a successor of ID3. It normalizes information gain using a "split information" and the ratio is called a gain ratio. The attribute A with the maximal value of $GainRatio(A)$ is selected as the splitting attribute.

If an attribute A is continuous, it is necessary to determine "appropriate" diving value, so-called split-point, that divides continuous values into two groups. At first, the values of the attribute A are ordered to create an increasing sequence: $a_1 < a_2 < \ldots < a_i < a_{i+1} < \ldots < a_v$. Then "appropriate" diving value can be calculated using the equation

$$splitpoint_i = \frac{a_i + a_{i+1}}{2}. \tag{3.15}$$

The split-point is always a midpoint between two adjacent values. Successively, the node creation is simulated for each $split\text{-}point_i$ as is shown in Figure 3.6 and $H_{Ai}(S)$ is calculated. Finally, the $split\text{-}point_i$ that has minimal $H_{Ai}(S)$ is selected. On the basis of chosen $split\text{-}point_i$, the discretization into two groups can be performed: $A \leq split\text{-}point_i$ and $A > split\text{-}point_i$.

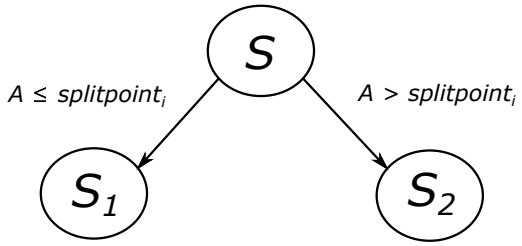

Figure 3.6: The node creation during the split-point determination

The decision tree pruning is a technique to reduce the size of decision trees. Pruning reduces the complexity of the final classifier and improves predictive accuracy by the reduction of overfitting. The first one is the so-called prepruning method in which the tree-building process is stopped earlier to prevent overfitting. The second one is the so-called postpruning method in which the tree is created first and then the branches and levels of the decision tree are reduced. Both methods have advantages and disadvantages. The postpruning method is more reliable then prepruning method, however, it is computational more demanding because of a whole decision tree generation. Therefore both methods are used for decision tree pruning.

The decision tree classifiers are suitable for exploratory knowledge discovery because any domain knowledge or parameter setting is not needed for the construction of a decision tree. Moreover, decision trees can deal with multidimensional data and the representation of the knowledge using tree form is easily understandable by humans. In general, the decision tree classifiers are accurate and classification accuracy on unknown data can be improved using tree pruning. On the other hand, their accuracy can be affected by noisy data or outliers contained in the training data. The next disadvantage is scalability because it is effective only if the training data is stored in memory.

Random forest is an extension of a decision tree. The extension uses many decision trees to classify an object. The class to which an object is classified is selected on the basis of majority value which is a result of individual decision trees.

3.2.2 Naive Bayes

Bayesian classifiers are statistical classifiers based on Bayes' theorem that predict probabilities of class membership, for example, the probability that a particular sample falls into a certain class.

Naive Bayesian classifiers suppose that given attributes are independent on each other. It results in the simplification of required calculations, therefore the classifiers are called "naive". Let X be a certain data sample $X = \{x_1, \ldots, x_n\}$ to be included in one of the classes $C_1, \ldots, C_m$. It is classified to the class C_i, where $P(C_i|X)$ is maximal. The conditional probability that sample X belongs to class C_i supposing known attribute values of X can be calculated as

$$P(C_i|X) = \frac{P(X|C_i)P(C_i)}{P(X)}\,. \tag{3.16}$$

Because $P(X)$ is constant, it is needed to find the maximal value of nominator. The probability that a data sample is classified to the class C_i is defined as

$$P(C_i) = \frac{|s_i|}{|S|}\,, \tag{3.17}$$

where s_i is a set of training samples of class C_i and S is a set of all training samples. The term $P(X|C_i)$ can be expressed using

$$P(X|C_i) = \prod_{k=1}^{n} P(x_k|C_i)\,. \tag{3.18}$$

Recall that attribute values are conditionally independent of one another. It is necessary to determine a type of each attribute. If an attribute is categorical, $P(x_k|C_i)$ is defined as

$$P(x_k|C_i) = \frac{|s_{ik}|}{|s_i|}\,, \tag{3.19}$$

where s_{ik} is a set of training samples of the class C_i where the value of k^{th} attribute equals to x_k. In the case of a continuous attribute, it is supposed to have a Gaussian distribution with a mean μ and standard deviation ρ

$$g(x, \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}\,, \tag{3.20}$$

and then

$$P(x_k|C_i) = g(x_k, \mu_{C_i}, \sigma_{C_i})\,. \tag{3.21}$$

If $P(x_k|C_i) = 0$ holds for some k, then $P(X|C_i) = 0$. The Laplacian correction avoids computing zero probability value by adding one sample to each set.

Studies have found that the performance of Naive Bayesian classifier, decision tree, and selected neural networks are comparable. Bayesian classifiers are very accurate and fast for large databases. Nevertheless, the attribute independence is assumed and an increase in error rate can appear due to the little amount of available probability data.

3.2.3 Bayesian Network

Unlike Naive Bayesian classifiers, Bayesian networks (belief networks, probabilistic networks) allow to define class conditional independencies between subsets of variables. Bayesian network consists of two parts – a directed acyclic graph and a set of conditional probability tables. Each node of directed acyclic graphs corresponds with a random discrete or continuous variable. The variables represent actual data attributes or "hidden variables" that relates to the level of user knowledge, certain beliefs, general interests, and present goals. A probabilistic dependence is represented by an arc. A directed arc from a node A to a node B means that A is a parent of B, in other words, B is a descendant of A. Individual nodes represent attributes and edges express dependencies between attributes. Each node in a Bayesian network has one conditional probability table (CPT) which defines the conditional distribution $P(Y|Parents(Y))$, where $Parents(Y)$ represents the parents of Y. Let $X = (x_1, \ldots, x_n)$ be a data sample specified by attributes $Y_1, \ldots, Y_n$. The joint probability distribution is defined as follows

$$P(x_1, \ldots, x_n) = \prod_{i=1}^{n} P(x_i|Parents(Y_i)), \tag{3.22}$$

where $P(x_1, \ldots, x_n)$ expresses the probability of a combination of certain values of X and $P(x_i|Parents(Y_i))$ can be determined using the records in the conditional probability table for Y_i. The result of the classification process can be a single class label or a probability distribution giving the probability of each class.

The Bayesian networks enable attribute dependency and provide a graphical model of causal relationships that can be used during a learning phase. On the other hand, the networks are computationally intensive because the probability of any branch in the network has to be calculated.

3.2.4 Neural Network

The multilayer feed-forward network (see Figure 3.7) is the type of fully-connected neural network where each output of a unit is used as an input of each unit in the next layer.

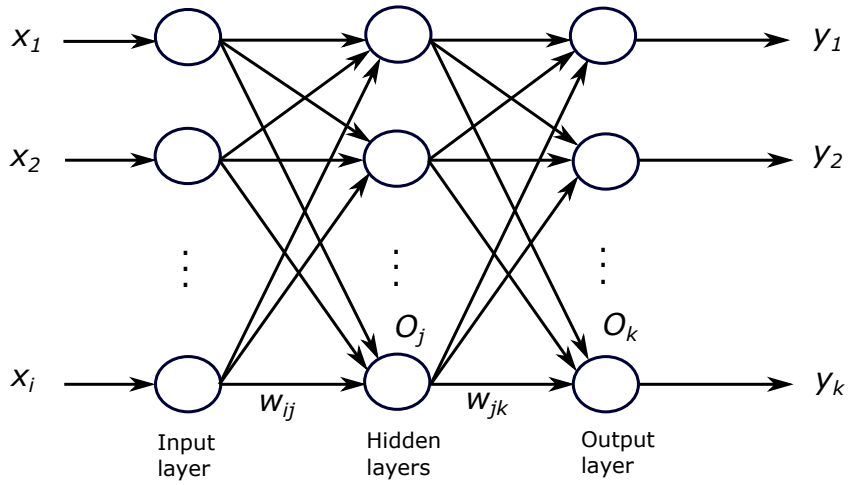

Figure 3.7: Multilayer feed-forward neural network

It uses the backpropagation algorithm (see Algorithm 2) that modifies the weights iteratively during the learning phase to correctly predict the class label of the input samples.

Algorithm 2 Neural network learning for classification or numeric prediction, using the backpropagation algorithm [8].

Input:

- D – a data set consisting of the training tuples and their associated target values,
- l – the learning rate,
- *network* – a multilayer feed-forward network.

Output: A trained neural network.

Method:

Initialize all weights and biases in *network*;
while terminating condition is not satisfied **do**
    for each training tuple X in D **do**
        for each input layer unit j **do**
            $O_j = I_j$;
        end for
        for each hidden or output layer unit j **do**
            $I_j = \sum_i w_{ij} O_i + \theta_j$;
            $O_j = \frac{1}{1+e^{-I_j}}$;
        end for
        for each unit j in the output layer **do**
            $Err_j = O_j(1 - O_j)(T_j - O_j)$;
        end for
        for each unit j in the hidden layers, from the last to the first hidden layer **do**
            $Err_j = O_j(1 - O_j) \sum_k Err_k w_{jk}$;
        end for
        for each weight w_{ij} in *network* **do**
            $\Delta w_{ij} = (l) Err_j O_i$;
            $w_{ij} = w_{ij} + \Delta w_{ij}$;
        end for
        for each bias θ_j in *network* **do**
            $\Delta \theta_j = (l) Err_j$;
            $\theta_j = \theta_j + \Delta \theta_j$;
        end for
    end for
end while

It is called feed-forward because connections between the units do not form a cycle. The multilayer feed-forward network is composed of an input layer, one or more hidden layers and an output layer. All layers consist of units, which are denoted as input units in the input layer, neurodes or output units in the hidden layers and output layer. The inputs, represented by the attributes of each sample, enter simultaneously the units in the input layer, then they are weighted and propagated towards one or more hidden layers.

The output layer determines, in the case of the classifier, the class for a given sample. The multilayer feed-forward network consisting of more than one hidden layer is called deep neural network. Learning techniques in deep neural networks enable to train much deeper and larger networks. For many problems, the performance of deep neural networks is significantly better than the performance of networks including only one hidden layer.

Backpropagation algorithm iteratively processes training samples and compares the prediction of network and the actual known target value which represents the class label of the training sample. The aim is to minimize the mean-square error between the prediction of the network and the actual known target value by weight modifications. Modifications are performed from the output layer up to the first hidden layer. The learning process continues until the weights start to converge.

Firstly, the algorithm ensures the initialization of weights and biases to small arbitrary numbers. Then the processing of each training sample includes the following two parts. In the first part, the training sample enters the input layer of the network and passes through it unmodified. Then the inputs and outputs of all units in the hidden and output layers are calculated. The net input of a unit j in a hidden or output layer is calculated as

$$I_j = \sum_i w_{ij} O_i + \theta_j \,, \tag{3.23}$$

where w_{ij} is the connection weight from unit i in the preceding layer to unit j, O_i is the output of unit i from the preceding layer, and θ_j is the unit bias that represents a threshold influencing the unit activity (see Figure 3.8).

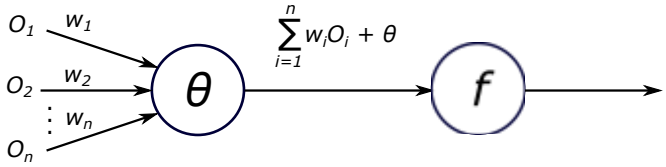

Figure 3.8: Activity of one neuron

The net input of each unit in the hidden and output layers is taken and then an activation function is applied to it. The function represents the activation of the neuron (the unit) and can be represented using the logistic or sigmoid function. The output of unit j, O_j, is defined as

$$O_j = \frac{1}{1 + e^{-I_j}} \,, \tag{3.24}$$

where I_j is the net input of a unit j in a hidden or output layer.

The second part of the algorithm corresponds with the error backpropagation that reflects the error of the network's prediction using modifying the weights and biases. The error Err_j for a unit j in the output layer is

$$Err_j = O_j(1 - O_j)(T_j - O_j) \,, \tag{3.25}$$

where O_j is the actual unit output j and T_j represents the known target value of the training sample.

The error of a hidden layer unit j is calculated as

$$Err_j = O_j(1 - O_j) + \sum_k Err_k w_{jk}\,, \tag{3.26}$$

where w_{jk} is the connection weight from unit j to a unit k in the following higher layer and Err_k is the error of unit k. The following equations are used for weight modification

$$\Delta w_{ij} = (l) Err_j O_i\,, \tag{3.27}$$
$$w_{ij} = w_{ij} + \Delta w_{ij}\,, \tag{3.28}$$

where Δw_{ij} represents the change of weight w_{ij}. The variable l, called the learning rate, is a constant with typical values from 0.0 to 1.0. It prevents getting stuck at a local minimum in decision space and helps to find the global minimum. It is recommended to set the learning rate to $\frac{1}{t}$, where t is the current number of iterations through the training set. Biases are modified using the equations

$$\Delta\theta_j = (l) Err_j\,, \tag{3.29}$$
$$\theta_j = \theta_j + \Delta\theta_j\,, \tag{3.30}$$

where $\Delta\theta_j$ is the change in bias θ_j. The approach when weights and biases are modified after processing of each sample is called case updating. If weights and biases are updated after all the samples in the training set have been processed, the approach is called epoch updating. In that case, one iteration through the training set is called an epoch. Terminating conditions that stop the training are:

- all Δw_{ij} in the precedent epoch are below a given threshold,
- the percentage of incorrectly classified samples in the precedent epoch is under a given threshold,
- a predefined number of epochs has been reached.

Neural networks are highly tolerant to noisy data, are able to classify patterns which were not included in the training set and do not require the analysis of relationships between attributes. Furthermore, neural networks are suitable for continuous attributes and are naturally parallel. The disadvantages are long training times, low interpretability and necessity to determine many parameters, for example, network topology.

3.2.5 Support Vector Machines

Support Vector Machines (SVM) is a method to classify both linear and non-linear data. The original training data can be transformed into a higher dimensional space using a non-linear mapping. The goal is to choose the linear hyperplane with a maximal margin around the separating hyperplane to divide samples into two classes. The hyperplane is found using support vectors (the subset of the training samples with the minimal distance from the hyperplane that have a major influence on its shape) and margins that are determined by the vectors (see Figure 3.9 where support vectors are marked in red colour). The SVM is able to classify objects into two classes.

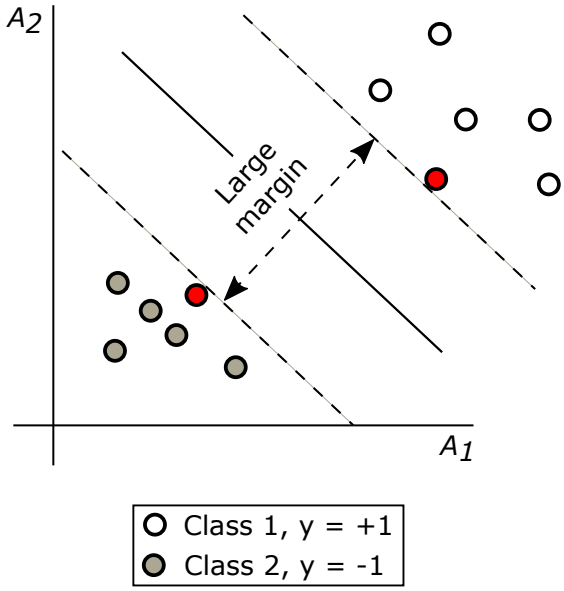

Figure 3.9: Maximum separating hyperplane and support vectors

A separating hyperplane can be defined using the equation

$$W \cdot X + b = 0, \tag{3.31}$$

where $W = \{w_1, w_2, ..., w_n\}$ is a weight vector, n is the number of attributes, and b is a scalar, usually called as a bias. Let us consider two input attributes, A_1 and A_2, as is shown in Figure 3.9. Training tuples are defined as $X = (x_1, x_2)$, where x_1 is the value of attribute A_1 and x_2 is the value of attribute A_2. If b is regarded as additional weight and marked as w_0, the equation (3.31) can be rewritten to the equation

$$w_0 + w_1 x_1 + w_2 x_2 = 0. \tag{3.32}$$

Weights can be adjusted to define the hyperplanes that represent the "sides" of the margin and are expressed using the following equations

$$H_1 : w_0 + w_1 x_1 + w_2 x_2 \geq 1 \quad for\ y_i \quad = +1, \tag{3.33}$$

$$H_2 : w_0 + w_1 x_1 + w_2 x_2 \leq -1 \quad for\ y_i \quad = -1. \tag{3.34}$$

A tuple satisfying the equation (3.33) belongs to class $+1$, and a tuple that meets the equation (3.34) belongs to class -1. Support vectors are training tuples that fall on hyperplanes H_1 or H_2 satisfy the following equation created using the equations (3.33) and (3.34)

$$y_i(w_0 + w_1 x_1 + w_2 x_2) \geq 1, \forall i. \tag{3.35}$$

The classification into more classes can be realized using a binary classifier for each class. The SVM provides good accuracy, are able to model complex nonlinear decision boundaries and are much less prone to overfitting than other methods. On the other hand, the training phase is very time-consuming.

3.2.6 K-Nearest Neighbours

Nearest-Neighbours classifier considers individual samples stored in an n-dimensional space where each sample represents one n-dimensional space point. A K-Nearest-Neighbours (KNN) classifier selects the closest K training samples to the unknown one. Euclidean distance (see Equation (3.14)) is used as a distance metric. The unknown sample is classified to the most frequent class among its K-Nearest Neighbours.

The K-Nearest-Neighbours classifiers are easy to implement. However, KNN has low accuracy supposing noisy or irrelevant attributes which can be improved by the attribute weighting and the pruning of noisy data samples. Looking for the K-nearest and the training phase are time-consuming.

3.3 Prediction and Clustering

The best-known method for prediction is simple linear and multiple linear regression. A lot of nonlinear problems can be transformed to the mentioned regression types. The simple linear regression expects data in the form $(x_1, y_1), (x_2, y_2), ..., (x_s, y_s)$, for $i = 1, \ldots, s$, where x_i represents an input attribute and y_i is an output attribute, whose value is predicted for new data. The data is approximated using the line defined by the equation $Y = aX + b$. It is necessary to determine the coefficients a and b. A frequently used method is called the method of least squares. Its goal is to find the coefficients a, b so that the sum of squares of the errors in Figure 3.10 is as low as possible.

Unlike simple linear regression where an output attribute depends only on one attribute, the multiple linear regression considers the dependency of an output attribute on more attributes. It expects data in the form $(x_{11}, x_{12}, \ldots, x_{1v}, y_1), (x_{21}, x_{22}, \ldots, x_{2v}, y_2), \ldots, (x_{s1}, x_{s2}, \ldots, x_{sv}, y_s)$. The approximation is performed using the equation

$$Y = a_0 + a_1 X_1 + a_2 X_2 + \ldots + a_v X_v \,, \tag{3.36}$$

where the coefficients $a_0, a_1, a_2, \ldots, a_v$ have to be determined. Some convenient classification methods like neural networks, K-Nearest Neighbours of Support Vector Machines can be also used for the prediction.

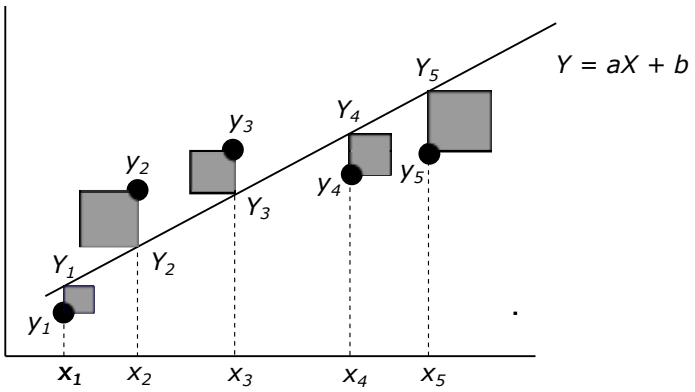

Figure 3.10: The method of least squares

Clustering is the process that divides objects (data samples) into clusters (groups) based on object similarity. Individual clusters contain similar objects that differ from the objects of other clusters. The object similarity is determined using the attribute values that describe

the objects and distance measures. The cluster analysis allows to find the data distribution and the characteristics of individual clusters. The clustering can be also used as a data preprocessing technique for classification algorithms.

The basic clustering methods can be divided into four categories: partitioning, hierarchical, density-based and grid-based. Partitioning methods are designed to find mutually exclusive clusters of spherical shape and are mostly distance-based. Cluster center can be calculated using mean or medoid of objects in that cluster. The methods are effective for small- or medium-size data sets. Hierarchical methods create a hierarchical decomposition of the set containing data objects. It is impossible to correct wrong splits and merges. Density-based methods are able to find clusters of arbitrary shape and to remove outliers. Created clusters are dense object areas in space separated using areas with low density. Cluster density means that a minimum number of points must be in "neighbourhood" of a given point. Grid-based methods divide the object space into a finite number of cells that create a grid structure. The main advantage of the methods is fast processing time which is often independent on the number of data objects, however, dependent on grid size.

A k-means is a well-known and commonly used partitioning method based on centroid-based technique that uses the cluster centroid (center point), C_i, to represent the cluster. It utilizes within-cluster variation to express the quality of a cluster that is defined as

$$E = \sum_{i=1}^{k} \sum_{p \in C_i} dist(p, c_i)^2, \tag{3.37}$$

where E is the sum of the squared error for objects in the data set, p is the point in space that represents a given object, c_i is the centroid of cluster C_i. The aim of that objective function is to make the final k clusters as compact and as separate as possible. The k-means algorithm defines the cluster centroid as the mean value of the points in the cluster. At first, it randomly selects k objects from data set D. The selected objects represent cluster centers. Each remaining object is assigned to the cluster based on the Euclidean distance between the object and the cluster center. Afterwards, the k-means algorithm gradually improves the within-cluster variation. It calculates the new mean of individual clusters using the objects that were assigned to the cluster in the preceding step. The objects are reassigned on the basis of the newly computed cluster centers. The algorithm continues until the clusters created in the current step are the same as the clusters formed in the preceding step.

Chameleon is a hierarchical clustering method that defines the similarity between clusters using dynamic modeling. It merges two clusters if the following conditions are met: cluster interconnectivity is high and clusters are close to each other. A k-nearest-neighbour graph approach is utilized to create a sparse graph, where a data object is represented by a vertex, and two vertices are connected using edge if one object belongs to the k-most similar objects to the other. The weight of the edge expresses how similar two objects are. Firstly, a graph partitioning algorithm is used to divide the k-nearest-neighbour graph into many small subclusters with the aim to minimize the edge cut. Then an agglomerative hierarchical clustering algorithm is utilized to merge the most similar subclusters.

The density of an object o (and similarly the density of the neighbourhood) is dependent on the number of objects that are close to o (are in the neighbourhood). DBCAN is a density-based clustering method that looks for objects with dense neighbourhoods (core objects). Clusters are represented by dense areas that include connected core objects and their neighbourhoods. It is necessary to define user-specified parameters.

The first one is $\epsilon > 0$ that defines a neighbourhood radius for every object. The second parameter is $MinPts$ that expresses the density of dense regions. If the ϵ-neighbourhood of the object includes at least $MinPts$ objects, then the object is called the core object.

CLIQUE is a grid-based method that enables to discover density-based clusters in subspaces. Firstly, it separates the d-dimensional data space into rectangular units that did not overlap. It finds dense cells in all subspaces. Then, CLIQUE uses the dense cells in each subspace to create clusters of arbitrary shape. The aim is to use the maximal regions to cover connected dense cells. A maximal region is a hyperrectangle in which every cell is dense and it is not possible to extend the region further in any dimension in the subspace. At first, an arbitrary dense cell is selected, then CLIQUE looks for the biggest region that covers the cell. Afterwards, it tries to cover the remaining dense cells. The procedure continues unless all dense cells are covered.

Chapter 4

Model Evaluation and Selection

The predictive accuracy of a classifier can be expressed using several evaluation metrics, for example accuracy, error rate, precision and recall. Cross-validation and Bootstrap methods are techniques used to assess accuracy on the basis of randomly sampled parts of the data. The "best" classifier can be selected using cost-benefit and receiver operating characteristic (ROC) curves. The chapter is based on [8, 42].

4.1 Classifier Performance Evaluation

Data samples can be divided into two parts. The positive samples belong to the main class of interest and the negative samples represent all other samples. True positives (TP) relate to the positive samples that the classifier labelled correctly. True negatives (TN) refer to the negative samples that were correctly labelled using the classifier. False positives (FP) are the negative samples that the classifier labelled incorrectly as positive. False negatives (FN) relates to the positive samples that were incorrectly labelled as negative.

Four previously described terms are included in a confusion matrix (see Table 4.1) that helps to analyze how well a given classifier can recognize samples of different classes.

		Predicted class		
		C_1	C_2	Total
Actual class	C_1	TP	FN	P
	C_2	FP	TN	N
	Total	P'	N'	P+N

Table 4.1: Confusion matrix

Let us consider m classes. Then the confusion matrix is a table consisting of at least m rows and m columns, where $m \geq 2$. Individual rows represent an actual class of a sample and columns represent a predicted class of the sample (or vice versa). An entry $CM_{i,j}$ in the first m rows and m columns denotes the number of samples of the class i that were mislabeled by the classifier as samples of the class j. The accuracy of the classifier is good if the samples are mostly situated along the main diagonal of the confusion matrix (i.e. the entries $CM_{i,i}$ for $i = 1, \ldots, m$ and the number of remaining samples is closed to zero, it means FP and FN are around zero. The table can be extended by additional rows and columns expressing totals. Number of samples of class C_1 is expressed using formula $P = TP + FN$, and number of samples of class C_2 can be calculated by equation $N = FP + TN$. Number

of samples labelled as positive is expressed using formula $P' = TP + FP$, and number of samples labelled as negative can be calculated by equation $N' = FN + TN$. Total number of samples is expressed as $TP + TN + FP + FN$, or $P + N$, or $P' + N'$.

Several measures can be used to evaluate classifier performance. The classifier accuracy (recognition rate) on a given test set represents the percentage of correctly classified samples by the classifier from the test set

$$accuracy = \frac{TP + TN}{P + N} . \tag{4.1}$$

The next evaluation measure is error rate (misclassification rate) of a classifier M that can be calculated as $1 - accuracy(M)$ or

$$error\ rate = \frac{FP + FN}{P + N} . \tag{4.2}$$

The sensitivity is related to the true positive (recognition) rate that is defined by equation

$$sensitivity = \frac{TP}{P} . \tag{4.3}$$

The specificity is the true negative rate computed as

$$specificity = \frac{TN}{N} . \tag{4.4}$$

The accuracy is a function of sensitivity and specificity

$$accuracy = sensitivity \frac{P}{(P + N)} + specificity \frac{N}{(P + N)} . \tag{4.5}$$

Precision expresses a measure of exactness

$$precision = \frac{TP}{TP + FP} . \tag{4.6}$$

The recall is a measure of completeness

$$recall = \frac{TP}{TP + FN} = \frac{TP}{P} . \tag{4.7}$$

Classifiers can be also compared according to the following criteria. Speed denotes the computational costs for generating and using classification rules. Robustness expresses correctness of predictions made by the classifier for noisy data or data containing missing values. Scalability is the ability to create the classifier effectively for a large amount of data. Interoperability expresses how difficult is to understand the classifier or predictor.

4.2 Model Testing

During the testing phase, the accuracy of the created model is determined. It is necessary to have enough testing data. The same data used for model training should not be used for model testing. It can cause misleading overoptimistic estimates because of over-specialization of the learning algorithm to the data. The correct approach for the

measurement of the model accuracy is to use a test set that consists of class-labelled samples which were not used during the training phase. The same ratio of classes as in the training set should be preserved.

In k-fold cross-validation, the data set is randomly divided into k mutually exclusive subsets ("folds"), $P_1, P_2, \ldots, P_k$, containing approximately the same number of samples. Training and testing phase is performed k times. The part P_i represents the test set in iteration i and all other parts are used for model training. It means that each sample is used in the same number of times for training and only once for testing purposes. The model accuracy can be expressed as the ratio of the total number of correctly classified samples in the k iterations to the total number of samples.

The special case of k-fold cross-validation is called leave-one-out where k is set to one. Consequently, the test set is represented by only one sample that is "left out" from samples. The stratified cross-validation ensures that the folds are stratified to preserve the same class distribution as in the data set. Stratified 10-fold cross-validation is generally recommended to estimate accuracy because of its relatively low bias and variance.

The bootstrap method selects samples for training randomly. If n samples are required for model training, the sampling with replacement is performed n times. As a result, the samples can appear in the training set multiple times. The unselected samples form the test set. On average, 63.2 % of samples are chosen for training and 36.8 % of samples is used for testing. In sampling without replacement, the training set includes 75 % of samples and 25 % of samples are in the testing set.

4.3 Classifier Comparison

The costs and benefits (risks and gains) related to the classification model are assessed using the true positives, true negatives, false positives, and false negatives. The cost related to a false positive is much lower than the cost of a false negative. The fact can be reflected by the setting of different cost. Analogically, the benefits related to a true positive and a true negative decision can differ.

Receiver operating characteristic curve (ROC) shows the trade-off between true positive rate (TPR) and the false positive rate (FPR). On the basis of a test set and a model, it is possible to determine TPR and FPR. TPR is the percentage of positive samples that are correctly labelled by the model. FPR is the percentage of negative samples that are mislabeled as positive. The classification model has to be able to determine the probability of the class predicted for each test sample. Then it is possible to rank and sort the samples. The sample with the highest probability of assigning to the positive class is placed at the top of the list. Similarly, the sample with the lowest probability of assigning to the positive class appears at the bottom of the list. The sample with the highest probability of assigning to the positive class is placed on the ROC curve near the centre of the coordinate system. The accuracy of the classification model corresponds with the area under the ROC curve (see Figure 4.1). The accuracy of the first model (denoted in red colour) is higher than the accuracy of the second one (marked in green colour).

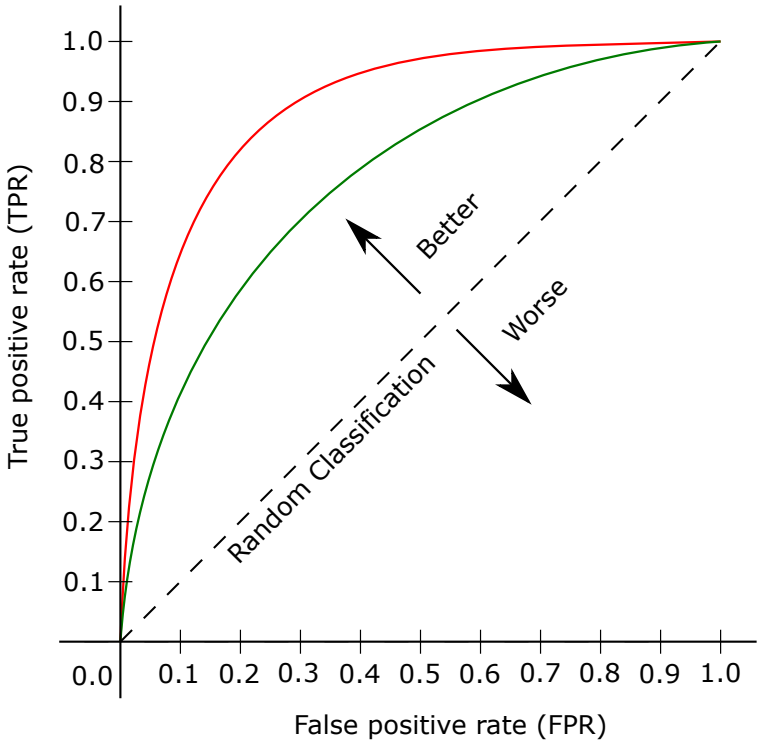

Figure 4.1: ROC curves

Chapter 5

Test Environment

The BeeeOn system is an open source project founded by the Faculty of Information Technology, Brno University of Technology. The project's main goal is to develop the system that connects devices from different manufacturers and allows to handle all devices in a uniform manner. It is possible to join sensors using various wireless communication protocols (i.e. Z-Wave, Bluetooth, IQRF, VPT, etc.). The server and the gateway are the main parts of the system architecture.

5.1 Hardware Platform

The BeeeOn gateway is based on the open source hardware board A10-OLinuXino-LIME by Olimex with the A10 1GHz Cortex-A8 ARMv7. This board is extended by the BeeeOn PAN coordinator module with RF (Radio Frequency) capability for wireless communication that includes a pressure sensor (see Figure 5.1).

Figure 5.1: The OlimexA10-OLinuXino-LIME with the BeeeOn PAN coordinator

The BeeeOn gateway has to be connected to the Internet via Ethernet cable so that communication with the server was possible. It has to stay awake continuously to communicate with devices which wake up from sleep mode. Therefore a power adaptor is used for constant power supply [21]. The BeeeOn Gateway application is platform independent,

however, some libraries are necessary for successful compilation. It receives measured data from sensors and converts it to the format specified for a given protocol. Then the data is stored on the server and it is accessible using REST API [22].

Sensors measure data and send it to the gateway. The communication between sensors and gateway is ensured by a USB interface. Sensors which communicate using the same protocol use given hardware units that implement the USB interface. BeeeOn sensors (see Figure 5.2) communicate with the gateway using the BeeeOn PAN coordinator module, Jablotron sensors (see Figure 5.3) use the USB interface Turris Dongle [38, 22].

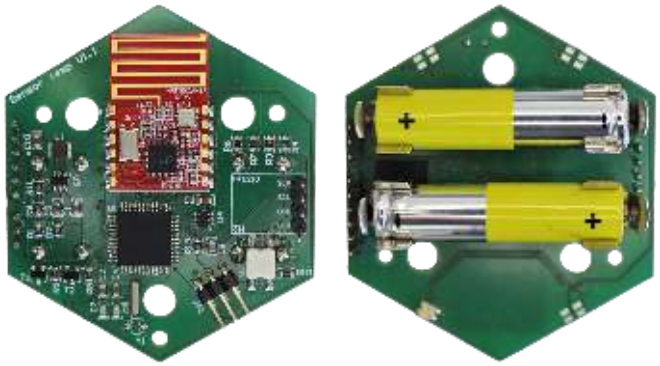

Figure 5.2: The BeeeOn sensor v1.2 – humidity and temperature sensor

Figure 5.3: The Jablotron sensor JA-83M – wireless magnetic window or door opening detector

The communication between the BeeeOn sensor and the BeeeOn PAN coordinator module is realized using the radio module MRF89XA in version MRF89XAM8A operating in the frequency range 863–870 MHz. Devices operated in the 863 MHz frequency band. The radio module supports data rates up to 200 kbps, transmit output powers up to 13 dBm and maximum range of about 100 meters. SPI (Serial Peripheral Interface) is used as a communication interface between the microcontroller and radio module. It also serves for data transfer among devices [22].

The BeeeOn sensor v1.2 uses two AAA batteries and can measure temperature and humidity (using humidity and temperature sensor HTU21/SHT21). It is also possible to connect an external temperature sensor. The temperature accuracy of ±0.5 °C from −10 °C to 85 °C and humidity accuracy of ±2 % from 0 % to 100 % [22].

The Jablotron sensor JA-83M is designed to detect the opening of doors, windows, etc. It is battery-powered and communicates with other devices wirelessly using a radio protocol. It operates in the frequency of 868 MHz [13, 22].

It was necessary to select appropriate relative humidity and temperature sensors. Except for the BeeeOn sensor, two other sensors were considered. The Protronix combined sensor NLII-CO2+RH+T-IQRF measures temperature, relative humidity, and carbon dioxide concentration. The temperature accuracy of ±0.4 °C from 0 °C to 40 °C and humidity accuracy of ±3 % from 20 % to 80 % and ±6 % from 0 % to 100 % [25]. The IQ Home SN-THC-02 sensor also measures carbon dioxide concentration, temperature, and relative humidity. The typical temperature accuracy of ±0.3 °C from 0 °C to 50 °C and typical humidity accuracy of ±3 % from 0 % to 85 % [12]. The temperature accuracy of all sensors is similar, but the Protronix sensor is less accurate if the relative humidity is measured in the range from 0 % to 100 %. Only the BeeeOn sensor supports the negative temperature measurement, therefore it was chosen for long-term measurement [22].

5.2 Measurement Scenarios

Devices supported by the BeeeOn system were used for measurement purposes. The BeeeOn sensors v1.2 were used for temperature and humidity measurement, both indoor and outdoor. Jablotron sensor served for detection of window opening and closing. The BeeeOn gateway received the data from sensors and forwarded it directly to the server for further analysis. The goal of the measurement in the flat in the suburb (see subsection 5.2.1) was to gather the data for further analysis, verify the relationship between the length of natural ventilation and the indoor humidity decrease, and determine the influence of key parameters on the indoor humidity. The key parameters include outdoor temperature and humidity, indoor temperature, and moisture sources. The secondary parameters are the airspeed, the weather type. The measured data in the room in the residence hall, in flat and in the office was used for testing purposes.

5.2.1 Flat in the Suburb

The measurement was performed in the flat that is situated in the calm city distinct of Brno-Řečkovice in the Czech Republic. The measurement was performed in the bedroom of volume 53.8 m^3 (see Figure 5.4) and in the bathroom of volume 9.4 m^3 (see Figure 5.5).

The bedroom has a wooden floor, two doors and one window. The window frame material is plastic and the window padding is a double glass. The window width is 0.5 m and height is 1.4 m. The window is in height of 0.9 m. It faces east and therefore the bedroom can be warmed by sunshine till noon. The radiator of the central heating is located below the window.

The possible source of humidity could be a presence of a tomcat or 1–3 people, drying of clothes (mostly from one to six pieces hanging on a clothes valet). Taking a shower in a near bathroom or natural ventilation in the other rooms could produce further moisture. The reason is that all rooms have an old wooden door, slightly deformed and hard to close. Therefore the airflow through a door could occur. The last possible moisture source can be one cactus that is grown in the bedroom.

The BeeeOn sensors and Jablotron sensor were placed in the bedroom. One BeeeOn sensor (marked in red colour) was located on the interior parapet close to the unopenable casement (about 0.5 m) in the height of 0.9 m. It was found out that the sensor was affected

by the heating. Therefore, the second BeeeOn sensor (marked also in red colour) was placed on the small table that is in front of a window (approximately 1.2 m) in the height of 0.5 m. It is essential to denote that only data measured using the second sensor was used for data analysis. The third BeeeOn sensor (marked in green colour) was located on the exterior parapet. All sensors were sheltered using a thick cardboard paper to prevent excessive warming caused by direct sunlight. The Jablotron sensor (marked in blue colour) which enabled to detect both full and partial opening of the window was fixed on the upper part of the casement.

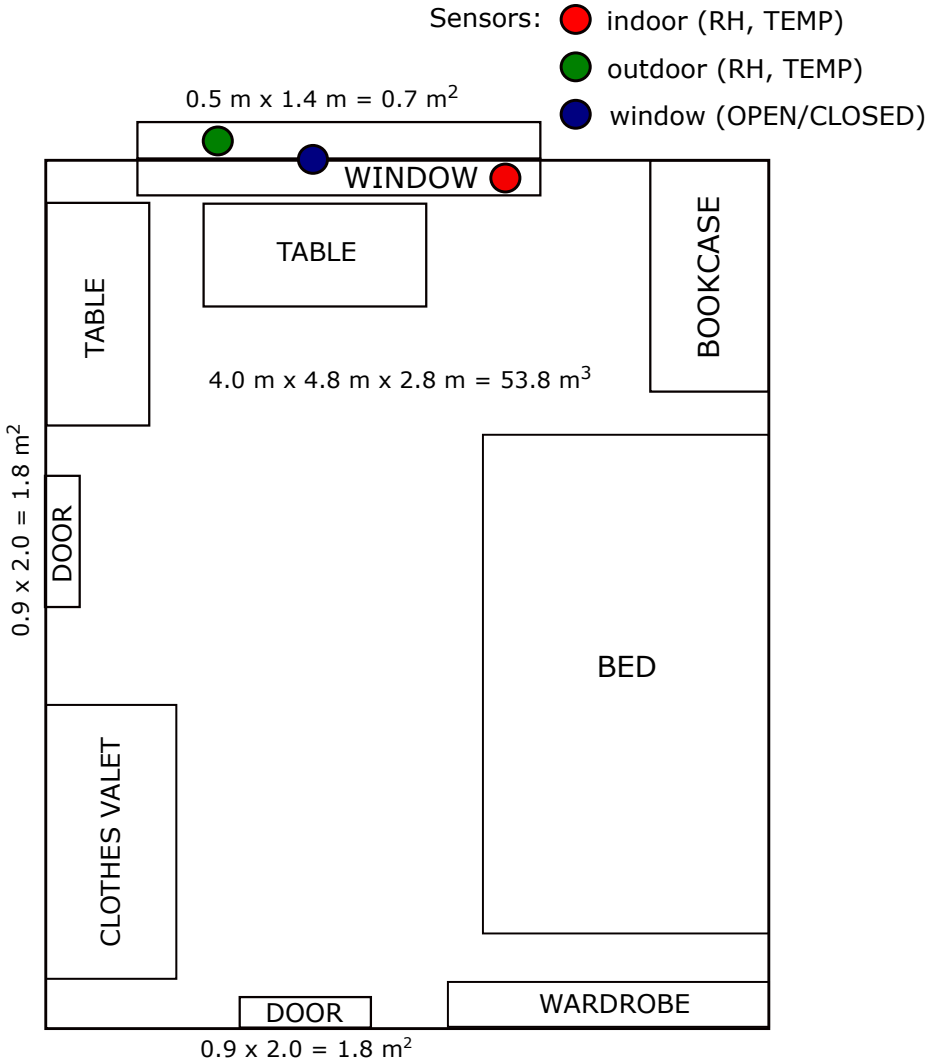

Figure 5.4: Bedroom plan

The bathroom has a tiled floor, one small window with a wooden frame and double glass padding and one door. The window width is 0.5 m, and the height is 0.9 m. It leads to airshaft that is formed between two adjacent houses. The main source of humidity is showering, washing machine, and drying of wet towels. The BeeeOn sensor (marked in red colour) was placed on the right wall in height of 1.65 m.

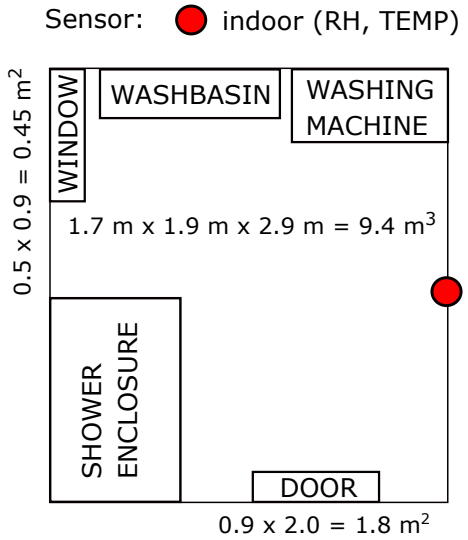

Figure 5.5: Bathroom plan

The measurement in the bedroom and in the bathroom was performed continuously from 20 September 2018 to 20 April 2019, and from 18 July 2018 to 19 April 2019, respectively. Only natural ventilation was performed. During all measurements, relative humidity and temperature were measured by the BeeeOn sensors and sent to the gateway every 15 seconds. In the bedroom, the window was wide open and both doors were closed. The other parameters were manually observed. The airspeed values were divided into three groups (calm, moderate, strong) and five types of weather were defined (shine, partly cloudy, cloudy, rain, snow). The window opening and closing were recorded both automatically by the Jablotron sensor and manually. In the bathroom, the window and door were closed during showering. The showering events had to be manually noted.

5.2.2 Room in Residence Hall

The residence hall room for three people is situated in Brno-Královo pole in the Czech Republic. The measurement lasted seven months (from 7 October 2018 to 29 April 2019). The width of the room is 3.1 m, the length is 6.5 m and the height is 2.7 m. The volume of the room is approximately 54.4 m^3. The room has one door and one double-wing window. The width of the one wing is 0.9 m and the height is 1.3 m. The window is in the height of 0.9 m. It faces northeast, therefore the room can be influenced by sunshine till noon. Two indoor sensors and one outdoor sensor were used to measure relative humidity and temperature. The first indoor sensor was placed in the height of approximately 1.8 m on the shelf and the outdoor sensor was situated on the exterior parapet. The second indoor sensor was placed on the window casement to detect both full and partial opening.

5.2.3 Room in the Flat

The single room is situated in Brno city centre in the Czech Republic. The measurement lasted one month (from 3 April 2019 to 29 April 2019). The width of the room is 5 m, the length is 7 m and the height is 2.5 m. The volume of the room is approximately 87.5 m^3. The room has one window and one door. The width of the window is 0.7 m and the height is 1.6 m. The first sensor was placed on the wardrobe in the height of 1.8 m and it measured

relative humidity and temperature. The second one was placed on the window casement to detect both full and partial opening.

5.2.4 Office

The measurement was performed for one month (from 1 April 2019 to 29 April 2019) in the office located in Brno-Královo pole in the Czech Republic. The volume of the office is 115 m^3. The office has 2 wooden windows with wooden frames and one wooden door. Only one of the windows was opened. The window width is 0.5 m and height is 1.6 m. The door was often opened or partially closed. Therefore, the measurement could be affected by the neighbouring room, where hourly meetings (6–10 people) were sometimes held.The other possible source of humidity can be a presence of 3–4 people during the working days and maximally 2 people during the weekend. The relative humidity and temperature sensor was placed in the center of the office (in distance about 2.8 m from the window, in a height of 1.5 m) to avoid direct sunlight.

Chapter 6

Design of Detectors and Predictor

The hardware platform and measurement scenarios were described in previous chapter 5. The first aim of the thesis was to evaluate measured data used to design a system detecting events related to a humidity change. Window opening (see section 6.2) and shower event (see section 6.3) were selected for further analysis. The next aim was to define a proper way how to regulate air change rate. Therefore, the optimal ventilation length prediction was designed (see section 6.4).

It was necessary to choose a suitable data mining tool for data analysis. Nowadays, many data mining tools are available, but the choice of the best one is not simple. The most used tools for data mining are RapidMiner, Orange, Weka, and KNIME [26, 14]. RapidMiner offers user-friendly integration of the latest and established data mining techniques. Moreover, it is the only tool which is language independent, it can be implemented on any system, and furthermore, it integrates maximum algorithms of all mentioned data mining tools. It is possible to create and run data mining tasks (processes) using a graphical user interface that enables to design a process using drag and drop of operators, setting parameters, and combining operators. Therefore, RapidMiner was used for data analysis [27].

6.1 Attribute Description

All measured data used for calculation of individual attributes was stored in the database. The assumption was that all sensors measured (no interruptions occurred) at the moment of event detection. Values of indoor and outdoor relative humidity were also expressed and stored as absolute and specific humidity (see Equations (2.9) and (2.11)). The relative humidity was converted to the specific humidity to eliminate the dependency on temperature. Moreover, the change in specific humidity level was more significant than the change in relative humidity and therefore, the specific humidity was more suitable for further data analysis. The decrease or increase of specific humidity level is influenced by room volume, area of an openable part of the window, door state (opened/closed), heating, plants, drying laundry or ventilation in other rooms. The distance of the sensor from the window or the location of a room where the measurement is performed also affects the specific humidity level. Each event was described using measured and derived attributes (see Table 6.1 and Table 6.2).

Attribute Description	Unit	Domain
measurement date and time (in YYYY-MM-DD HH:MM:SS)	–	numeric
indoor relative humidity	[%]	numeric
indoor absolute humidity	[g·m^{-3}]	numeric
indoor specific humidity	[g·kg^{-1}]	numeric
outdoor relative humidity	[%]	numeric
outdoor absolute humidity	[g·m^{-3}]	numeric
outdoor specific humidity	[g·kg^{-1}]	numeric
indoor temperature	[°C]	numeric
outdoor temperature	[°C]	numeric
window action	–	discrete {opening, closing, nothing}
shower action	–	discrete {showering, nothing}

Table 6.1: Description of measured attributes

Attribute Description	Unit	Domain
difference between quantity values[a]	[%] – relative humidity [g·m^{-3}] – absolute humidity [g·kg^{-1}] – specific humidity [°C] – temperature	numeric
first difference not only successive values		
first difference only successive values		
second difference only successive values		
normalized first or second difference		
difference between measured quantity values and quantity values of linearized course		
geometric mean of differences		
arithmetic mean of differences		
variance of differences		
standard deviation of differences		
number of positive differences	-	

(a) Attributes for window opening and shower detection

[a] The attribute was used only for window opening detection.

Attribute Description	Unit	Domain
difference between quantity values	[%] – relative humidity [g·m^{-3}] – absolute humidity [g·kg^{-1}] – specific humidity [°C] – temperature	numeric
quantity values after linearization		
difference between quantity values after linearization		
distance between a data point and cluster trendline	-	
distance between a data point and cluster centroid		

(b) Attributes for optimal ventilation length prediction

Table 6.2: Derived attribute description

Table 6.2a contains attributes for window opening and shower detection (see section 6.2 and section 6.3), Table 6.2b includes attributes for optimal ventilation length prediction (see section 6.4). Description of derived attributes follows:

- the first difference, not only successive values – the difference between the value of a quantity measured at the time of window opening and value measured in a time point, several seconds before or after the window was open,
- the first difference, only successive values – the difference between values of a quantity measured in various time points, several seconds before or after the window was open,
- the second difference, only successive values – the difference between first difference values of a quantity in various time points, several seconds before or after the window was open,
- normalized first or second difference – difference between values of a quantity measured in various time points, several seconds before or after the window was open divided by difference of two given time points,
- difference between measured quantity values and quantity values of linearized course, several seconds before or after the window was open,
- difference between quantity values – the difference between values of a quantity measured indoor and outdoor in various time points, several seconds before or after the window was open,
- quantity values after linearization – time interval of given quantity values before window opening and after window closing is linearised, then the linearised values of given quantity at the moment of window opening and closing are chosen,
- difference between quantity values after linearization (see the previous attribute),
- the geometric mean of differences – geometric mean of values within a given time interval,
- the arithmetic mean of differences – arithmetic mean of values within a given time interval,

- variance of differences – variance of values within a given time interval,
- the standard deviation of differences – standard deviation of values within a given time interval,
- number of positive differences – the number of positive differences in various time points, several seconds before or after the window was open,
- distance between a data point and cluster trendline (see section 6.4),
- distance between a data point and cluster centroid (see section 6.4).

The calculation of differences is demonstrated on typical specific humidity decrease and is shown in Figures 6.1 and 6.2. The red line represents the window opening event in time point t_e. The sampling interval is denoted as h. The number of considered time points before and after the window opening event is denoted as m and n, respectively. In figures below, the window opening event occurred in time point $t_e = 36{:}45$, the sampling interval $h = 15$ seconds and $m = n = 3$.

The principle of calculation for differences between not only successive values is shown in Figure 6.1. The forward differences are calculated by subtracting values in time points $t_e - (t_e + nh)$ and backward differences are calculated as $t_e - (t_e - mh)$. The approach allows to use various distances between considered time points. On the other hand, the calculation of differences between only successive values considers only neighbouring time points (see Figure 6.2). The forward differences are calculated as $[t_e + (n-1)h] - (t_e + nh)$ and backward differences as $(t_e - mh) - [t_e - (m-1)h]$.

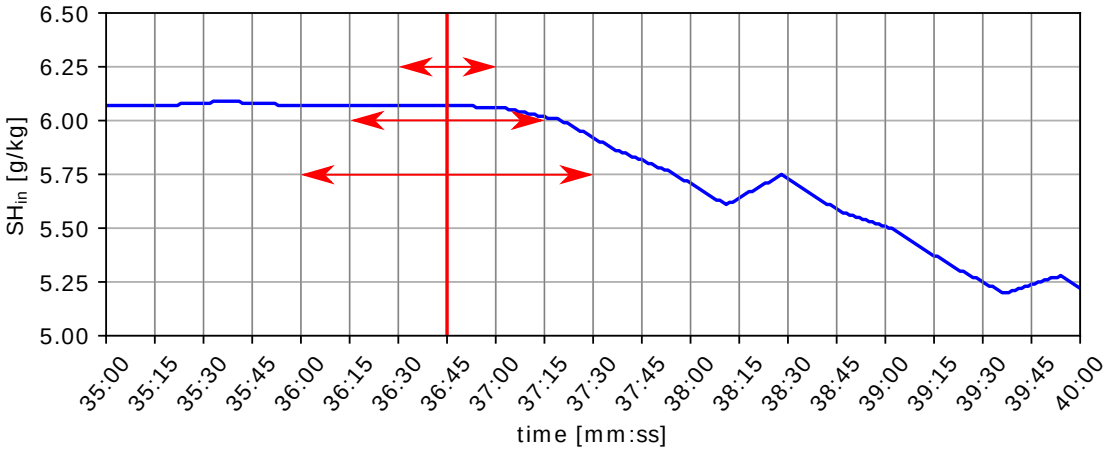

Figure 6.1: Differences between not only successive values

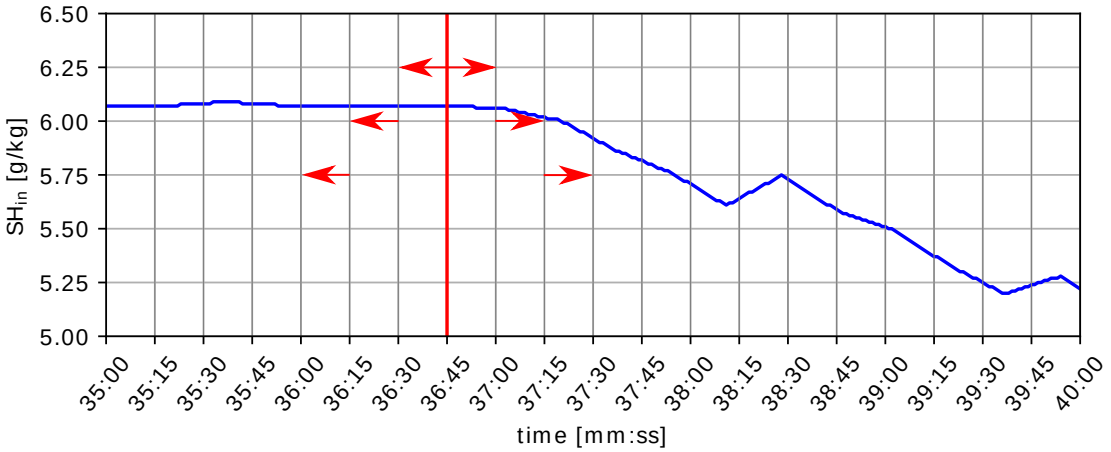

Figure 6.2: Differences between only successive values

6.2 Window Opening Detection

The aim was to detect window opening based on a decrease of specific humidity level. Training set contained derived attributes (see Table 6.2a) calculated in the moment of window opening and several seconds before window opening to equilibrate the number of samples in both classes, *opening* and *nothing* (see *window action* attribute in Table 6.1). Derived attributes calculated using values of specific, absolute humidity, and temperature were considered. The positive or negative value of difference indicates the growth or decrease of a given quantity and also the speed of change. Differences were calculated in various time points, several seconds before and after the window opening. The number of samples after the window opening significantly influenced the model accuracy and delay of window opening detection. Therefore, the considered number of samples was as low as possible to reduce undesired delay and keep required accuracy. Testing set included derived attributes calculated during given time interval.

First of all, it was necessary to download measured data at the moment of window openings from the database to create the training set. The training set contained 998 records, 499 records corresponded with the moment of the window opening (class *opening*), the rest of records was measured before window opening (class *nothing*). The attributes were calculated in the interval of 10 minutes before and 3 minutes after the window opening event, with the time step of several seconds. Total number of attributes was 1 354.

Then data was sampled using a sampling interval of 30 seconds. In real application the mentioned interval is sufficient for event detection and it also decreases the power consumption of sensors. The data, measured during four months (from 26 November 2018 to 26 March 2019), was downloaded from database to create testing set. In total, testing set consisted of 244 849 records.

Classification using chosen classification methods (see section 3.2) was performed using RapidMiner. Process for window opening detection is shown in Figure 6.3. At first, it was necessary to load CSV files, where training and testing data sets were stored. Then *datetime* attribute in both data sets was set as an identifier and *window action* attribute in both data sets was set as a label (class). Mentioned steps were performed using subprocess operator *Read training data* and *Read testing data*. Data included in the training set was classified using SVM (*SVM* operator) and afterwards, the trained model was used on the

testing set (*Apply Model* operator). The goal was to predict a class for unknown data. Data was evaluated using *Evaluation* operator.

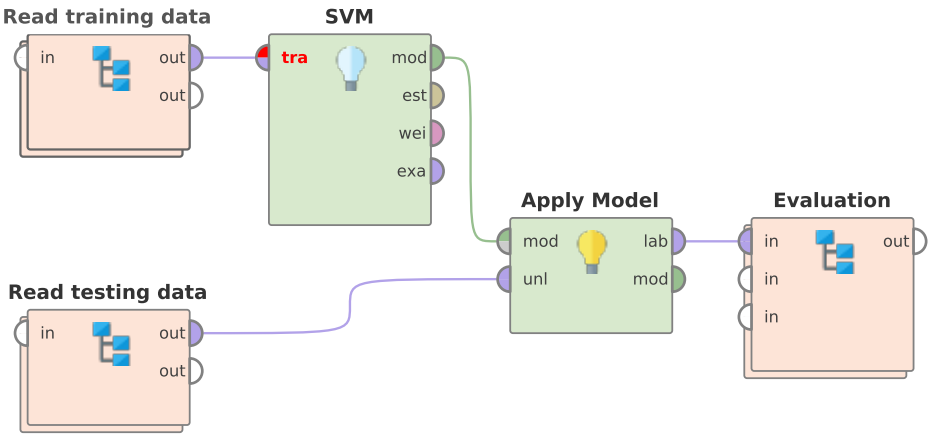

Figure 6.3: RapidMiner process for window opening detection

One of the outputs of the process was the accuracy of a given classification method (*standard accuracy*). The assumption was that the window was not opened in a given time interval before and after the detected window opening. False positives that fell into this interval were considered as true negatives. As a consequence, the accuracy was increased. Accuracy calculated using above-mentioned approach was called as *accuracy with tolerance*. False positives that occurred 10 minutes before and after window opening were not taken into account.

Table 6.3 represents accuracy, number of true positives, false negatives, true negatives, and false positives obtained by selected classification methods.

Classification Method	**Accuracy Type**	**Accuracy [%]**	**TP**	**FN**	**TN**	**FP**
Decision Tree	standard	93.64	439	4	228 843	15 563
	with tolerance	97.31	441	2	237 828	6 578
Random Forest	standard	92.98	437	6	227 224	17 182
	with tolerance	96.90	440	3	236 827	7 579
Naive Bayes	standard	87.93	431	12	214 856	29 550
	with tolerance	92.22	439	4	225 361	19 045
Neural Network	standard	93.66	438	5	228 893	15 513
	with tolerance	97.29	443	0	237 773	6 633
Deep Learning	standard	92.74	436	7	226 634	17 772
	with tolerance	96.72	440	3	236 380	8 026
SVM	standard	94.86	420	23	231 852	12 554
	with tolerance	**98.52**	439	4	240 789	3 617
*K*NN	standard	71.75	402	41	175 277	69 129
	with tolerance	76.16	443	0	186 022	58 384

Table 6.3: Accuracy of selected classification methods

The best accuracy was reached by Support Vector Machines (SVM) that was used for the iterative process of model training. The aim of model training was to reach the best possible accuracy and to eliminate incorrectly detected window openings (false positives) that could be caused by the factors mentioned in section 6.1. Therefore, in each iteration, 10 records corresponding to the class *nothing* (see *window action* attribute in Table 6.1) in training set were replaced with false positives from testing set to train the model. The maximum number of replaced records was equal to the number of records containing class *nothing* to keep equilibrium. After 41 iterations, 350 records were replaced in the training set. Accuracy, number of true positives, false negatives, true negatives, and false positives obtained using SVM after the 1^{st} iteration are shown in Table 6.4.

Classification Method	**Accuracy Type**	**Accuracy [%]**	**TP**	**FN**	**TN**	**FP**
SVM	standard	94.87	419	24	231 873	12 533
	with tolerance	98.59	438	5	240 964	3 442

Table 6.4: Accuracy of SVM after the 1^{st} iteration

The results after the 41^{st} iteration are represented in Table 6.5. In general, accuracy of the model gradually improved during individual iterations. The number of false positives was reduced from 3 617 to 0, while the number of false negatives increased from 4 to 39. In real applications, it is more important to eliminate the number of false positives, i.e. incorrectly detected window opening events.

Classification Method	**Accuracy Type**	**Accuracy [%]**	**TP**	**FN**	**TN**	**FP**
SVM	standard	98.23	344	99	240 164	4242
	with tolerance	99.98	404	39	244 406	0

Table 6.5: Accuracy of SVM after the 41^{st} iteration

Due to the limitation applied for the number of replaced records, it was needed to manually preprocess data contained in the testing set. The goal was to remove significant specific humidity decreases that occurred even though the window in the room where the measurement was performed was closed. In general, taking into consideration the daily regimen of occupants and time when decreases were measured, it seems that decreases were caused by window openings in the rooms where the measurement was not performed.

With the aim to evaluate the generality of created and trained detector, it was used on testing sets containing data measured in three other rooms (see subsections 5.2.2, 5.2.3 and 5.2.4). Achieved accuracy, number of true positives, false negatives, true negatives, and false positives are presented in Tables 6.6 – 6.8.

Results obtained using data measured in the room in the residence hall (see subsection 5.2.2) are shown in Table 6.6. Measurement was performed approximately seven months (from 7 October 2018 to 29 April 2019). The testing set comprised of 432 655 records. The window was opened 759 times and window opening event was correctly detected 634 times. The 31 false positives were probably caused by different parameters of the room in the residence hall and the room where the detector was trained (see subsection 5.2.1). The volume of both rooms was approximately same, but the window in room in the residence hall was larger and the measurement was not affected by ventilation in another room.

Classification Method	Accuracy Type	Accuracy [%]	TP	FN	TN	FP
SVM	standard	98.45	268	491	425 696	6 200
	with tolerance	99.96	634	125	431 865	31

Table 6.6: Accuracy of SVM in the room (residence hall)

Table 6.7 shows results obtained after measurement evaluation in the room described in subsection 5.2.3. Testing set contained 39 744 records that were measured during one month (from 3 April 2019 to 29 April 2019). The number of window opening events was 22, from which only 9 events were correctly detected. The data analysis showed, that the change in specific humidity level after window opening was slower than the course used to train the detector. It probably caused 13 false negatives, but no false positives were detected.

Classification Method	Accuracy Type	Accuracy [%]	TP	FN	TN	FP
SVM	standard	99.85	0	22	39 686	36
	with tolerance	99.97	9	13	39 722	0

Table 6.7: Accuracy of SVM in the room (flat)

Table 6.8 represents results acquired after evaluation of data measured in the office (see subsection 5.2.4). Data was also gathered for one month (from 1 April 2019 to 29 April 2019). The testing set consisted of 37 750 records including 10 window openings. The detector was able to correctly detect 5 window opening events. The change in specific humidity level after window opening was also slower than the original change used to train the detector which could cause false negatives. The number of false positives could be affected by the exchange of air from the next room, because the door was open or only partially closed.

Classification Method	Accuracy Type	Accuracy [%]	TP	FN	TN	FP
SVM	standard	99.88	1	9	37 704	36
	with tolerance	99.96	5	5	37 731	9

Table 6.8: Accuracy of SVM in the office

It was found out that the created detector is not general. It is necessary to train it for a given room because decrease of specific humidity after window opening differs. A detector accuracy could be improved by the iterative process of detector training.

To improve time and memory consumption, the subset of attributes was selected using a metric called gain ratio (see equation (3.7)). The 15 attributes with the maximum gain ratio were selected from the 1 354 attributes (see Table 6.9). The derived attributes considering specific humidity (SH) and absolute humidity (AH) reached the highest gain ratio. It was found out that the values of the attributes taken in several seconds after the window opening event have a significant influence on the successful event detection. As a result, it was confirmed that a certain delay is necessary for the detection. The training set for the

window opening detector contained the same records as the training set obtained after the iterative process of training, but each record was described using only 15 attributes.

Attribute Description	Quantity	Number of seconds after event [s]	Gain Ratio
first difference not only successive values	AH	180	0.347
	SH	180	0.338
	AH	165	0.311
	SH	165	0.310
	SH	150	0.303
	AH	150	0.296
first difference only successive values	AH	125	0.285
second difference only successive values	AH	125	0.285
first difference only successive values	SH	125	0.283
second difference only successive values	SH	125	0.283
first difference not only successive values	AH	135	0.275
first difference not only successive values	SH	135	0.272
arithmetic mean of first differences only successive values	AH	0	0.246
first difference not only successive values	SH	120	0.245
first difference only successive values	AH	180	0.244

Table 6.9: Selected attributes based on the gain ratio

The execution time was calculated as an average of 10 runs. With all attributes, the average time was 3 021 milliseconds, while considering a reduced set of attributes, the average time was 14 milliseconds. The size of the training set was reduced from 8.6 MB (1 354 attributes) to 0.12 MB (15 attributes). The original accuracy of SVM was 99.98 %, however, after the selection it was moderately lowered to 99.97 % (see Table 6.10). The number of false negatives was almost the same, while the number of false positives increased from 0 to 41. The reached results showed the trade-off between time and memory consumption and the accuracy of SVM.

Classification Method	Accuracy Type	Accuracy [%]	TP	FN	TN	FP
SVM	standard	98.19	340	103	240 078	4 328
	with tolerance	99.97	401	42	244 365	41

Table 6.10: Accuracy of SVM after the attribute selection

6.3 Shower Detection

The goal was to detect showering based on the increase of specific humidity level. The task was similar to the window opening detection (see section 6.2). The training set included almost the same derived attributes as for window opening detection (see Table 6.2a). However, more time points after the beginning of showering were taken into account to ensure humidity increase detection even though showering lasted only about 2 minutes. The training set contained data measured when showering started and also when no showering event occurred to equilibrate the number of samples in *showering* and *nothing* classes. The testing set was also created using the derived attributes measured during a given time period.

Data measured at the beginning of shower events was downloaded from the database and utilized to create training set which was comprised of 1 234 records, 617 records were measured during showering, 617 records were obtained when no showering occurred. The attributes were calculated in the interval of 10 minutes before and 5 minutes after the showering event, with the time step of several seconds. Total number of attributes was 1 423. The testing set was defined using downloaded data sampled using a sampling interval of 30 seconds and measured during two months (from 1 November 2018 to 31 December 2018). It contained 149 383 records. The classification was performed using selected methods for classification in RapidMiner. The process for shower detection is the same as the process for window opening detection. Except for *standard accuracy*, *accuracy with tolerance* was also considered to eliminate false positives. The assumption was that next showering did not occur in a defined time interval before and after a detected one.

Table 6.11 shows accuracy, number of true positives, false negatives, true negatives, and false positives reached using chosen classification methods. The method with the highest accuracy and the performed iterative process were the same as in the case of window opening detection (see section 6.2 for details). Tolerance interval was set to 10 minutes which means that incorrectly detected showering events 5 minutes before or after correctly detected one were ignored.

Classification Method	**Accuracy Type**	**Accuracy [%]**	**TP**	**FN**	**TN**	**FP**
Decision Tree	standard	95.92	148	4	143 206	6 096
	with tolerance	97.35	151	1	145 341	3 961
Random Forest	standard	97.41	148	4	145 438	3 864
	with tolerance	98.73	151	1	147 408	1 894
Naive Bayes	standard	91.10	140	12	136 016	13 286
	with tolerance	92.46	151	1	138 033	11 269
Neural Network	standard	95.99	152	0	143 315	5 987
	with tolerance	97.39	152	0	145 396	3 906
Deep Learning	standard	95.70	152	0	142 874	6 428
	with tolerance	96.90	152	0	144 674	4 628
SVM	standard	97.74	149	3	145 922	3 380
	with tolerance	**99.01**	151	1	147 826	1 476
*K*NN	standard	92.00	146	6	137 354	11 948
	with tolerance	93.45	152	0	139 518	9 784

Table 6.11: Accuracy of selected classification methods

Accuracy, number of true positives, false negatives, true negatives, and false positives acquired after the 1^{st} and the 62^{nd} iteration are shown in Table 6.12 and Table 6.13, respectively. Generally, the classifier accuracy increased after the individual iterations. After the 62^{nd} iteration, the number of detected false positives was decreased from 1476 to 38, while the number of false negatives was increased from 1 to only 13. More iterations could not be performed because all records of the class *nothing* were replaced.

Classification Method	Accuracy Type	Accuracy [%]	TP	FN	TN	FP
SVM	standard	97.88	149	3	146 143	3 159
	with tolerance	99.14	151	1	148 025	1 277

Table 6.12: Accuracy of SVM after the 1^{st} iteration

Classification Method	Accuracy Type	Accuracy [%]	TP	FN	TN	FP
SVM	standard	99.41	98	52	148 404	829
	with tolerance	99.97	137	13	149 195	38

Table 6.13: Accuracy of SVM after the 62^{nd} iteration

Testing data was analysed to find out reasons why all false positives were not removed. The number of false positives was decreased from 38 to 15 when tolerance interval was increased from 5 to 10 minutes. It means that showering incorrectly detected 10 minutes before or after its start was ignored. Remaining false positives were detected in the case of temperature increase. Possible causes could be: window opening and closing, hair blow-drying, washing, door opening and closing. However, it was not possible to notice all the mentioned actions.

6.4 Optimal Ventilation Length Prediction

The goal of the task was to predict optimal ventilation length to decrease indoor specific humidity to the required level and to regulate the air change rate. Specific humidity can be also expressed as relative humidity (see Equation (2.12)) that is familiar to people. Data set contained measured data in the moment of window openings during seven months (from 20 September 2018 to 20 April 2019). In total, it contained 260 records. It was divided into a training set (70 % of data) where the ratio of individual classes was preserved and testing set (30 % of data). First, the window opening intervals were arbitrary with the aim to gather enough data and determine the dependency of the humidity changes on the measured quantities.

Nevertheless, the dependency of humidity changes on the measured quantities could not be derived because of the continuous character of the quantities. It was not possible to gather enough data to predict ventilation length with precision in seconds. Moreover, the precision in seconds is not necessary in the target domain. After data analysis, ventilation length intervals of 5, 10, 15, 20, and 25 minutes were selected to define the data model. Furthermore, specific humidity decrease during ventilation of 20 and 25 minutes differed less than shorter ventilation lengths. As a result, ventilation intervals longer more than 25 minutes were not considered.

The purpose of the model is to determine the ideal ventilation length based on the required indoor specific humidity decrease. During ventilation, the indoor specific humidity decrease is considerably affected by the outside specific humidity. In general, the higher difference between indoor and outdoor specific humidity is, the higher decrease of indoor specific humidity is observed. Figure 6.4 and Figure 6.5 express the dependency of decrease of indoor specific humidity on the difference between indoor and outdoor specific humidities. Data obtained during a given ventilation length forms one cluster. Figure 6.4 shows that data especially from clusters representing ventilation lengths of 15, 20, and 25 minutes strongly overlap. Therefore only the ventilation lengths of 5, 10, and 25 minutes were selected (see Figure 6.5).

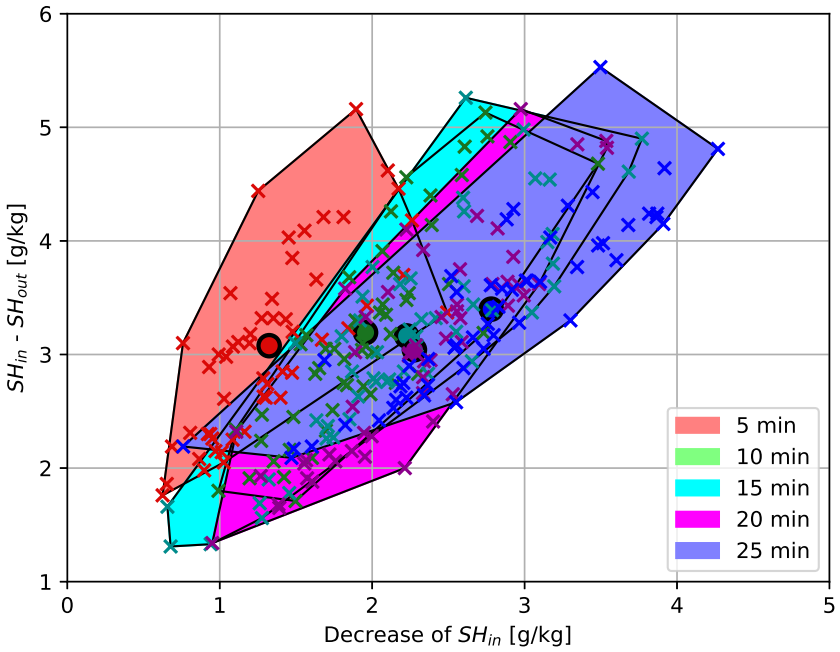

Figure 6.4: Dependency of decrease of indoor specific humidity on the difference between indoor and outdoor specific humidity for the ventilation intervals of 5, 10, 15, 20, 25 minutes

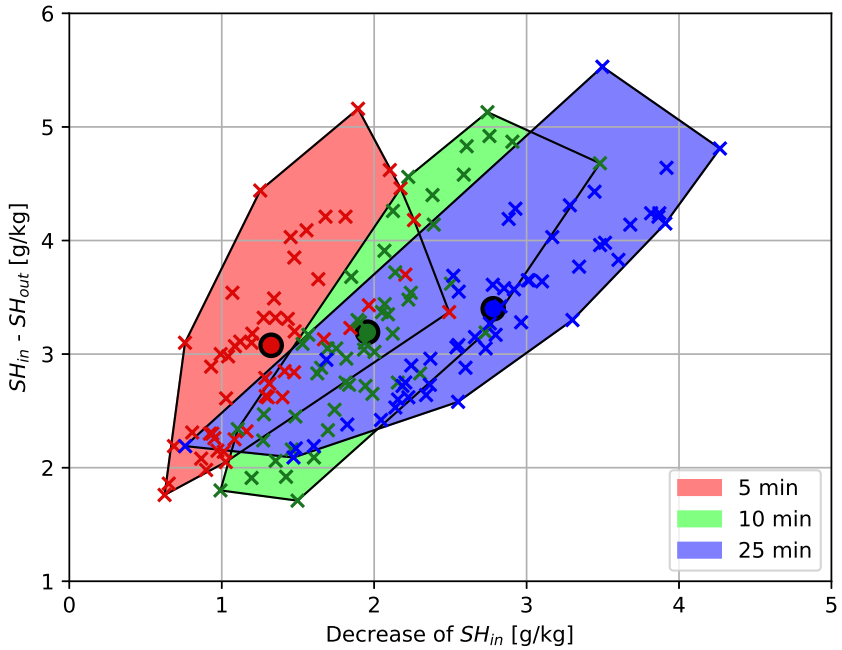

Figure 6.5: Dependency of decrease of indoor specific humidity on the difference between indoor and outdoor specific humidity for the ventilation intervals of 5, 10, 25 minutes

The dependency of interval length on the indoor specific humidity decrease was confirmed in cold months (from November 2018 to March 2019). The longer the interval is, the higher the indoor specific humidity decrease tends to be.

Prediction of the optimal ventilation length was based on the different attributes shown in Table 6.14. Attributes were divided into three groups. Measured attributes were obtained using operations with measured data. Calculated attributes were determined using the created model and analytic geometry approaches. All attributes included both, measured and calculated attributes. Prediction using measured attributes is straightforward because no model has to be created. However, a huge amount of different data is required for prediction. Moreover, data can be influenced by unpredictable events during window opening, for example, weather change or opening of the door. The main advantage of calculated attributes is their ability to reflect the dependency of specific humidity decrease on the difference between indoor and outdoor specific humidity which results in a more accurate expression of specific humidity decrease in the room. Furthermore, the accuracy of prediction is not decreased even though the data set is small.

Attributes	**All**	**Calculated**	**Measured**
difference between quantity values	×		×
distance between a data point and cluster trendline	×	×	
distance between a data point and cluster centroid	×	×	

Table 6.14: Attributes for optimal ventilation length prediction

Distance between a given data point and a trendline can be calculated using different approaches. The first approach (trendline) calculates a trendline for data of given ventilation length (see Figure 6.6) using the least squares method (see section 3.3).

Another possibility (average trendline) is to determine the average slope of the line passing through the points $[0, 0]$ and $[x_j, k_j \cdot x_j]$, where x_j is the maximal decrease of indoor specific humidity within cluster j and k_j is the average slope of the lines passing through the points $[0, 0]$ and $[x_{ij}, k_{ij} \cdot x_{ij}]$, where x_{ij} is an x-axis of the data point i from cluster j and k_{ij} is a slope of the line (see Figure 6.7).

The third approach (trendline passing cluster centroid) uses k-means clustering to determine cluster centroids. Then the line passes through the points $[0, 0]$ and $[x_j, y_j]$, where $[x_j, y_j]$ is the centroid of the cluster j (see Figure 6.8).

Distance between given data point $A[x_{ij}, y_{ij}]$ and a cluster centroid $C[x_j, y_j]$ is calculated using the Euclidean distance formula

$$d_{AC} = \sqrt{(x_{ij} - x_j)^2 + (y_{ij} - y_j)^2}\,. \tag{6.1}$$

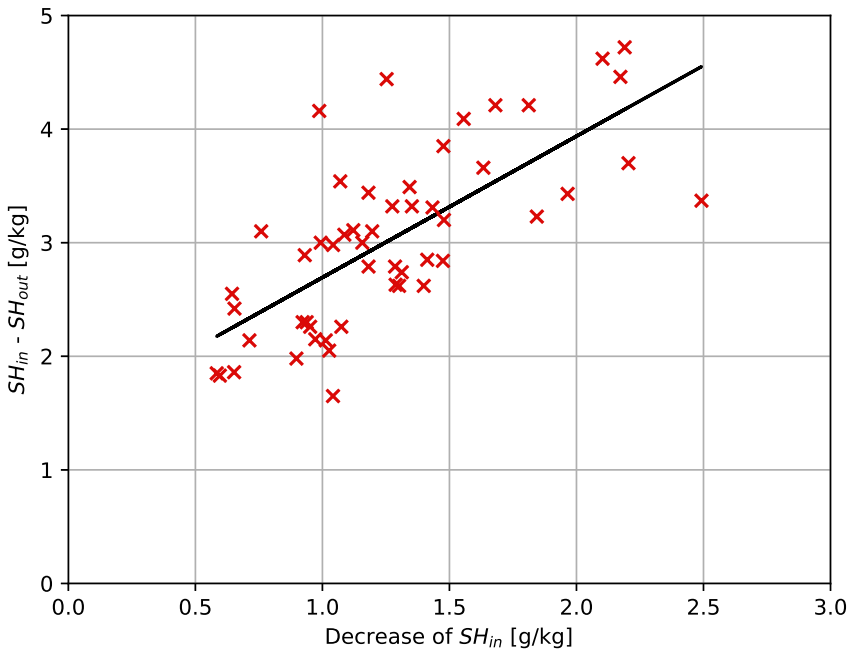

Figure 6.6: Trendline of a cluster calculated using least squares method

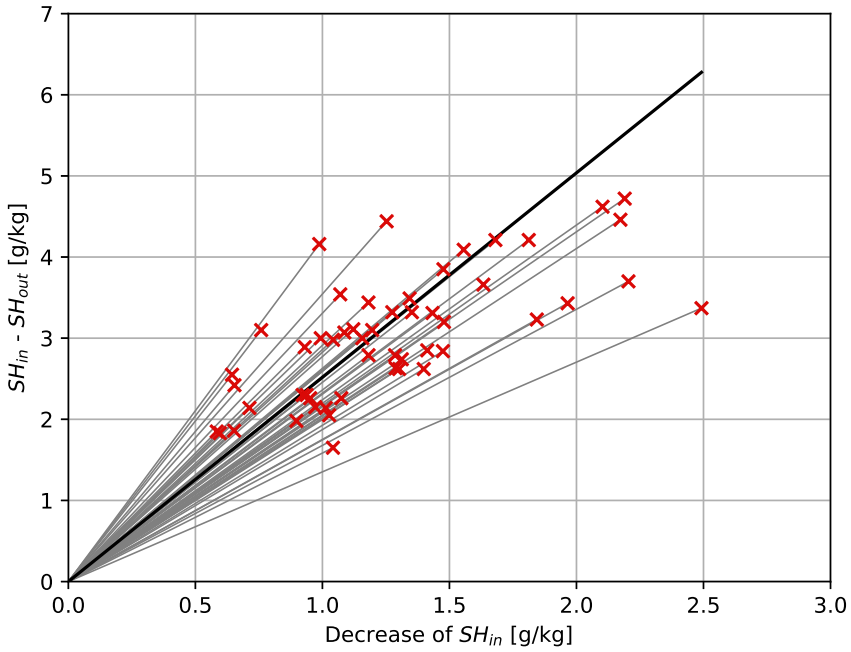

Figure 6.7: Average trendline of a cluster

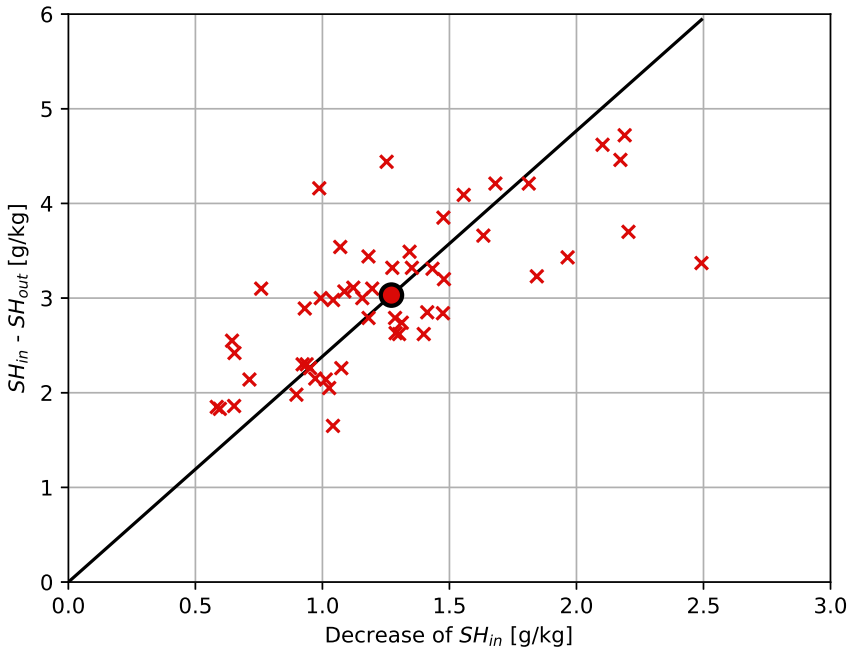

Figure 6.8: Trendline passing cluster centroid

First, measured data was downloaded from the database. Then data was split into training and testing set. Before prediction, it was needed to add attributes related to distance to training and testing set. Prediction was performed using decision tree because of its good interpretability and it was realized by RapidMiner.

Process for optimal ventilation length prediction is shown in Figure 6.9. Beginning of the process is the same as for window opening detection – *Read training data* and *Read testing data* subprocesses are used. It means that training and testing sets are loaded from CSV files, *datetime* attribute in the data sets is set as an identifier, and *ventilation_length* attribute in the data sets is set as a label (classes *5*, *10* and *25*). Moreover, the attributes used to create the model are removed (quantity value after linearization at the moment of window closing and the difference between quantity values after linearization at the moment of the window opening and closing). The decision tree is used for classification of data in training set (*Decision Tree* operator) and then trained model is applied on testing data set (*Apply Model* operator) with the aim to get a prediction on unknown data. Output of *Evaluation* operator are *standard accuracy* and a CSV file. Standard accuracy is calculated based on values in a confusion matrix and it expresses accuracy of classification using decision tree. However, it did not include knowledge about a target domain where selected clusters (data obtained during given ventilation intervals) were not completely disjunctive. Therefore, only half of the false positives that fell into the adjacent cluster was regarded as an error. Accuracy calculated using data in CSV file and the previously mentioned approach is called *modified accuracy*.

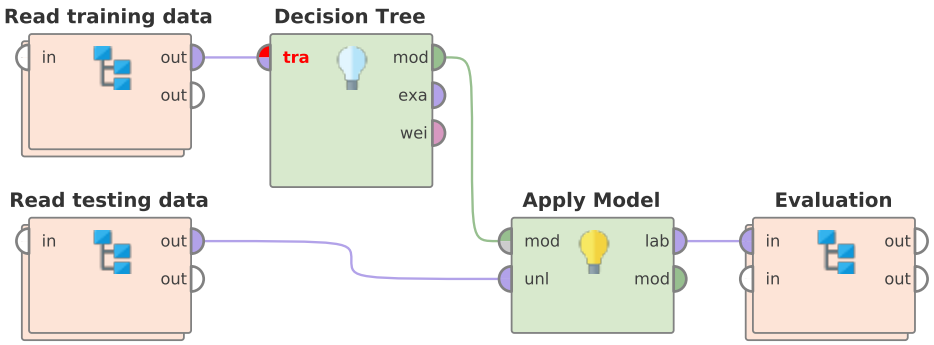

Figure 6.9: RapidMiner process for optimal ventilation length prediction

Results are summarized in Tables 6.15–6.23 that show the dependency of accuracy of decision tree on attributes and method of trendline calculation considering various differences between indoor and outdoor temperature in the moment of the window opening. Rows of tables correspond with selected attributes. For each set of attributes, standard and modified accuracy were calculated. Columns of tables denote previously mentioned approaches of a trendline calculation. In general, modified accuracy reached better results than standard accuracy for various intervals of temperature differences. The highest obtained accuracy for each interval of temperature differences is in bold. In the following text, modified accuracy is called accuracy in order to make the result description clearer. In training set, the classes always contained the same number of records.

Table 6.15 shows accuracy when all temperature differences were considered. The training set contained 156 records and the testing set included 104 records. The highest accu-

racy was reached using average trendline and trendline passing cluster centroid methods (87.02 %) for all attributes.

Attributes	Accuracy Type	Accuracy of Methods [%]		
		Trendline	Average Trendline	Trendline Passing Cluster Centroid
measured	standard	23.08		
	modified	40.87		
calculated	standard	64.42	70.19	68.27
	modified	78.85	85.10	83.65
all	standard	62.50	75.00	74.04
	modified	79.81	**87.02**	**87.02**

Table 6.15: Dependency of accuracy on attributes and method of trendline calculation, all temperature differences

The bigger difference between indoor and outdoor temperatures is, the more significant decrease of specific humidity is. Each ventilation interval was divided into several subintervals based on temperature differences which could result in the better data model accuracy. Following tables represent the accuracy of decision tree for data selected based on specified temperature differences that fell into the interval between 5 °C and 30 °C. It was necessary to divide the interval into subintervals of the same length. Furthermore, each subinterval had to contain at least 30 values to ensure valid data sample for classification. Therefore the subintervals were defined as follows:

- two subintervals: ⟨5.0 °C–17.5 °C), ⟨17.5 °C–30 °C),
- three subintervals: ⟨5 °C–13.3 °C), ⟨13.3 °C–21.6 °C), ⟨21.6 °C–30 °C),
- three subintervals: ⟨10 °C–15 °C), ⟨15 °C–20 °C), ⟨20 °C–25 °C).

Table 6.16 represents accuracy for temperature differences in the range from 5.0 °C to 17.5 °C. The training set was created using 96 records and the testing set comprised of 54 records. The highest accuracy was obtained by trendline passing cluster centroid method (88.89 %) for all attributes.

Attributes	Accuracy Type	Accuracy of Methods [%]		
		Trendline	Average Trendline	Trendline Passing Cluster Centroid
measured	standard	44.44		
	modified	61.11		
calculated	standard	61.11	74.07	74.07
	modified	80.56	87.04	86.11
all	standard	68.52	66.67	79.63
	modified	84.26	83.33	**88.89**

Table 6.16: Dependency of accuracy on attributes and method of trendline calculation, temperature differences: ⟨5.0 °C–17.5 °C)

Accuracy reached in the case of temperature differences from 17.5 °C to 30.0 °C is shown in Table 6.17. The training and testing set included 60 and 48 records, respectively. Trendline

passing cluster centroid method considering all attributes reached the highest accuracy (91.67 %).

Attributes	**Accuracy Type**	**Accuracy of Methods [%]**		
		Trendline	**Average Trendline**	**Trendline Passing Cluster Centroid**
measured	standard	37.50		
	modified	57.29		
calculated	standard	70.83	77.08	79.17
	modified	84.38	88.54	89.58
all	standard	68.75	81.25	83.33
	modified	81.25	90.62	**91.67**

Table 6.17: Dependency of accuracy on attributes and method of trendline calculation, temperature differences: ⟨17.5 °C–30.0 °C)

Table 6.18 shows accuracy for temperature differences ranging from 5.0 °C to 13.3 °C. The training set was created using 39 records and the testing set comprised from 26 records. The highest accuracy (90.38 %) was obtained using the trendline passing cluster centroid method for calculated attributes.

Attributes	**Accuracy Type**	**Accuracy of Methods [%]**		
		Trendline	**Average Trendline**	**Trendline Passing Cluster Centroid**
measured	standard	38.46		
	modified	65.38		
calculated	standard	57.69	80.77	84.62
	modified	76.92	88.46	**90.38**
all	standard	69.23	80.77	80.77
	modified	82.69	88.46	88.46

Table 6.18: Dependency of accuracy on attributes and method of trendline calculation, temperature differences: ⟨5.0 °C–13.3 °C)

Next interval of temperature differences was from 13.3 °C to 21.6 °C (see Table 6.19). The training set contained 87 records and the testing set included 74 records. The best accuracy (93.92 %) was reached by trendline passing cluster centroid method for calculated attributes.

Attributes	Accuracy Type	Accuracy of Methods [%]		
		Trendline	Average Trendline	Trendline Passing Cluster Centroid
measured	standard	18.92		
	modified	38.51		
calculated	standard	79.73	81.08	87.84
	modified	89.86	90.54	**93.92**
all	standard	74.32	83.78	83.78
	modified	87.16	91.89	91.89

Table 6.19: Dependency of accuracy on attributes and method of trendline calculation, temperature differences: ⟨13.3 °C–21.6 °C)

Acquired accuracy for temperature differences from 21.6 °C to 30.0 °C is presented in Table 6.20. Training and testing set contained 21 and 11 records, respectively. The highest accuracy was obtained using a trendline method (90.91 %) considering all attributes.

Attributes	Accuracy Type	Accuracy of Methods [%]		
		Trendline	Average Trendline	Trendline Passing Cluster Centroid
measured	standard	27.27		
	modified	40.91		
calculated	standard	72.73	63.64	63.64
	modified	86.36	81.82	81.82
all	standard	81.82	63.64	63.64
	modified	**90.91**	81.82	81.82

Table 6.20: Dependency of accuracy on attributes and method of trendline calculation, temperature differences: ⟨21.6 °C–30.0 °C)

Reached accuracy for temperature differences between 10.0 °C and 15.0 °C is in Table 6.21. The training set was created using 45 records and the testing set consisted of 24 records. The highest accuracy (85.42 %) was reached using the trendline method for all and calculated attributes and by average trendline method for all attributes.

Attributes	Accuracy Type	Accuracy of Methods [%]		
		Trendline	Average Trendline	Trendline Passing Cluster Centroid
measured	standard	29.17		
	modified	52.08		
calculated	standard	70.83	70.83	75.00
	modified	**85.42**	81.25	83.33
all	standard	75.00	75.00	75.00
	modified	**85.42**	**85.42**	83.33

Table 6.21: Dependency of accuracy on attributes and method of trendline calculation, temperature differences: ⟨10.0 °C–15.0 °C)

Table 6.22 shows accuracy for temperature differences ranging from 15.0 °C to 20.0 °C. The training set included 57 records and the testing set contained 41 records. Accuracy reached the highest value (98.78 %) in the case of average trendline method considering calculated attributes.

Attributes	**Accuracy Type**	**Accuracy of Methods [%]**		
		Trendline	**Average Trendline**	**Trendline Passing Cluster Centroid**
measured	standard	43.90		
	modified	64.63		
calculated	standard	70.73	97.56	87.80
	modified	85.37	**98.78**	93.90
all	standard	85.37	92.68	85.37
	modified	92.68	96.34	92.68

Table 6.22: Dependency of accuracy on attributes and method of trendline calculation, temperature differences: ⟨15.0 °C–20.0 °C)

The last interval of temperature differences was from 20.0 °C to 25.0 °C (see Table 6.23). The training set comprised of 33 records and the testing set included 21 records. The highest accuracy (95.24 %) was obtained by the average trendline method for all and calculated attributes and using the trendline passing cluster centroid method that considered all attributes.

Attributes	**Accuracy Type**	**Accuracy of Methods [%]**		
		Trendline	**Average Trendline**	**Trendline Passing Cluster Centroid**
measured	standard	14.29		
	modified	30.95		
calculated	standard	80.95	90.48	85.71
	modified	90.48	**95.24**	92.86
all	standard	80.95	90.48	90.48
	modified	90.48	**95.24**	**95.24**

Table 6.23: Dependency of accuracy on attributes and method of trendline calculation, temperature differences: ⟨20.0 °C–25.0 °C)

Optimal ventilation length prediction using calculated or all attributes was more accurate than prediction based only on measured attributes. Measured attributes were not able to reflect dependency of specific humidity decrease on the difference between indoor and outdoor specific humidity using a small amount of training data. The best obtained accuracy for individual intervals of temperature differences ranged from 85.42 % to 98.78 %. The accuracy was mostly reached using average trendline or trendline passing cluster centroid method. Accuracy of trendline method was the highest only in two cases when a small amount of data that met a required interval of temperature differences was available. A higher amount of data could improve the acquired results.

Chapter 7

Conclusion

The goal of this thesis was to deal with the high humidity level in buildings. All the goals of the work were completely fulfilled. At first, humidity issues were examined and major causes of excessive humidity were discussed together with possible solutions. The equations for humidity calculation and standards defining the comfort zone using temperature and humidity were explained.

The data mining methods were introduced with emphasis on classification methods. Further model evaluation and selection including classifier performance evaluation, model testing, and classifier comparison were discussed. The architecture of BeeeOn system was described together with five measurement scenarios. The long-term measurement of indoor and outdoor relative humidity and temperature was performed using a set of the BeeeOn sensors. The aim of the measurement was to gather enough data to detect events related to the humidity changes. The window opening and showering events were selected for further analysis.

Before data classification, it was necessary to define attributes describing the measured data. The detection of window opening event was based on indoor specific humidity decrease. Data measured at the moment of window opening for seven months (from 20 September 2018 to 20 April 2019) in the bedroom was used to train the detector. In total, the training data set was comprised of 834 records. Testing data set contained 244 849 records representing data gathered during four months (from 26 November 2018 to 26 March 2019). Several classification methods were considered for initial evaluation of detector accuracy. A lot of window openings were incorrectly detected (false positives) which was caused by window openings in the rooms where the measurement was not performed. To improve the accuracy, the iterative process of training was performed using the Support Vector Machines that reached the best accuracy in the initial evaluation. After 41^{st} iteration, all false positives were eliminated. The detector was tested on data measured in three other rooms. The generality of the detector was not confirmed due to various indoor specific humidity changes after window opening event. Therefore, the detector has to be trained for a given room. Time and memory consumption was improved by the attribute selection based on the metric called gain ratio. Considering 15 attributes with the highest gain ratio, the size of the training set was reduced from 8.6 MB (1 354 attributes) to 0.12 MB (15 attributes) and execution time of detector creation was decreased from 3 021 milliseconds to 14 milliseconds. On the contrary, the accuracy of the detector moderately decreased, while the number of false positives increased from 0 to 41.

Unlike the window opening detection, the detection of showering event used indoor specific humidity increase. The detector was trained using a data set consisting of 1 234 records

corresponding with data measured at the beginning of shower events. Testing data set contained 149 383 records representing data gathered during two months (from 1 November 2018 to 31 December 2018). Again, the iterative training of Support Vector Machines was performed. Nevertheless, 15 false positives were detected which could be caused by window opening and closing, hair blow-drying, washing, door opening and closing.

The regulation of air change was ensured by natural ventilation. The model for the optimal ventilation length prediction was based on indoor specific humidity decrease that depends on the difference between indoor and outdoor specific humidity. It was based on clusters defined by data measured during a given ventilation interval. The data was gathered for seven months (from 20 September 2018 to 20 April 2019). The selected ventilation intervals were $5, 10$ and 25 minutes to eliminate cluster overlapping. The attributes were divided into three groups: measured, calculated and all (measured and calculated). The measured attributes represented data measured by sensors. The calculated attributes used the distance between a given data point and a cluster trendline and distance between the data point and a cluster centroid. Cluster trendline was calculated using three different approaches. The training set was comprised of 156 records and the testing set contained 104 records. In both data sets, records described data measured at the moment of the window opening. With the aim to increase the accuracy of the ventilation length prediction, the data was divided into several subintervals based on differences between indoor and outdoor temperature. The obtained results showed that measured attributes are insufficient for the ventilation length prediction because a huge amount of data is necessary to define the dependency of specific humidity decrease on the difference between indoor and outdoor specific humidity. The reached accuracy was maximally 65.38 %. On the contrary, the calculated attributes considering the previously mentioned dependency reached a 20 % higher accuracy. The best approaches to calculate the cluster trendline were average trendline and trendline passing cluster centroid. The accuracy obtained using calculated attributes was not lower than 85.42 %.

All the presented results are based on data measured during winter. Several improvements can be done. The indoor specific humidity decrease after the window opening in summer is different because of the lower difference between indoor and outdoor specific humidity. As a result, the measurement in summer has to be performed. It was found out that the detector is not general. Therefore, the data annotation together with the iterative process of detector training could be automatized to create a system based on measured data in a certain room. The system could be used to detect the window opening. It could be also trained for multiple rooms using the automatized process. To predict the ventilation length, it is necessary to create a model based on measured data and select clusters which do not strongly overlap. Moreover, it is also possible to perform cluster analysis to find clusters of data characterizing another event, for example, a door opening. Afterwards, the detectors could be used for the detection of different events.

Bibliography

[1] Aggarwal, C. C.: *Data mining: the textbook*. Springer. 2015.

[2] Almalowi, S. J.: Gas Mixture and Air-conditioning. Online. 2016. Accessed on 2018-11-07.
Retrieved from:
https://www.taibahu.edu.sa/Pages/AR/DownloadCenter.aspx?SiteId=67101c3c-c3d1-4ae7-8a89-406568834388&FileId=3de8ca93-1cfb-4490-a6a9-7350a72ead16

[3] Bell, S.: A beginner's guide to humidity measurement. *NPL UK*. 2012.

[4] Boyle, B.: Humidity Generation and Humidity Measurement. Online. Accessed on 2018-11-17.
Retrieved from: http://info.owlstonenanotech.com/rs/owlstone/images/Humidity%20Generation%20and%20Humidity%20Measurement.pdf

[5] Determining Thermal Comfort Using a Humidity and Temperature Sensor. Online. May 2014. Accessed on 2018-11-02.
Retrieved from: https://www.azosensors.com/article.aspx?ArticleID=487

[6] Dry Bulb, Wet Bulb and Dew Point Temperatures. Online. 2004. Accessed on 2018-11-11.
Retrieved from:
https://www.engineeringtoolbox.com/dry-wet-bulb-dew-point-air-d_682.html

[7] Filippone, M.; Masulli, F.; Rovetta, S.; et al.: Input selection with mixed data sets: A simulated annealing wrapper approach. In *CISI'06-Conferenza Italiana Sistemi Intelligenti*. 2006.

[8] Han, J.; Pei, J.; Kamber, M.: *Data mining: concepts and techniques*. Elsevier. 2011.

[9] Hangar, P.: Aerographers Mate. Online. Accessed on 2018-11-29.
Retrieved from: http://meteorologytraining.tpub.com/14312/css/14312_31.htm

[10] Horr, Y. A.; Arif, M.; Katafygiotou, M.; et al.: Impact of indoor environmental quality on occupant well-being and comfort: A review of the literature. *International Journal of Sustainable Built Environment*. vol. 5, no. 1. 2016: pp. 1 – 11. ISSN 2212-6090.
Retrieved from:
http://www.sciencedirect.com/science/article/pii/S2212609016300140

[11] How to convert relative to absolute humidity. Online. August 2012. Accessed on 2018-12-01.
Retrieved from: https://carnotcycle.wordpress.com/2012/08/04/how-to-convert-relative-humidity-to-absolute-humidity/

[12] IQhome: IQ Home Sensor SN-xxx-02 Series Datasheet. Online. Accessed on 2018-11-10.
Retrieved from: https://www.iqrfalliance.org/product_files/sensor-family-datasheet-sn-xxx-02-.pdf

[13] Jablotron: *Jablotron JA-83M*. 2009.
Retrieved from:
https://www.jablotron.com/en/about-jablotron/downloads/?filename=ja-83m_en_mll51000.pdf&do=downloadFile

[14] Kadaru, B. B.; UmaMaheswararao, M.: An Overview of General Data Mining Tools. *International Research Journal of Engineering and Technology*. vol. 4, no. 9. 2017: pp. 930 – 936. ISSN 2395-0072.
Retrieved from: https://www.irjet.net/archives/V4/i9/IRJET-V4I9165.pdf

[15] Kohavi, R.; John, G. H.: Wrappers for feature subset selection. *Artificial intelligence*. vol. 97, no. 1-2. 1997: pp. 273–324.

[16] Kumar, V.; Minz, S.: Feature selection. *SmartCR*. vol. 4, no. 3. 2014: pp. 211–229.

[17] Kusiak, A.: Data Analysis: Models and Algorithms.

[18] Liu, H.; Motoda, H.: *Feature selection for knowledge discovery and data mining*. vol. 454. Springer Science & Business Media. 2012.

[19] Liu, H.; Yu, L.: Toward integrating feature selection algorithms for classification and clustering. *IEEE Transactions on Knowledge & Data Engineering*. vol. 17, no. 4. 2005: pp. 491–502.

[20] McDowall, R.: Chapter 3 - Thermal comfort. In *Fundamentals of HVAC Systems*, edited by R. McDowall. Oxford: Elsevier. 2007. ISBN 978-0-12-373998-8. pp. 34 – 44.
Retrieved from:
http://www.sciencedirect.com/science/article/pii/B9780123739988500030

[21] Nečasová, K.: *Extension of Wireless Sensor Protocol*. Bachelor's thesis. Brno University of Technology. Faculty of Information Technology. 2017.

[22] Nečasová, K.: *Analysis of Data to Solve Problems with Humidity in Buildings*. Project practice. Brno University of Technology. Faculty of Information Technology. 2018.

[23] NMTKQTU, R. N.; NMNKPOR, M. N.; VRKQSN, R. N.; et al.: Composition of Dry and Moist Air. 2002.

[24] Pavya, K.; Srinivasan, B.: Feature Selection Techniques in Data Mining: A Study. *International Journal of Science & Engineering Development Research*. vol. 2, no. 6. 2017: pp. 594–598.

[25] Protronix: Kombinované čidlo CO_2/RH/T s IQRF. Online. Accessed on 2018-12-18. Retrieved from: https://www.careforair.eu/data/pdf/CZ/DS/ds-NLII-CO2-RH-T-IQRF-cz.pdf

[26] Rangra, K.; Bansal, K.: Comparative Study of Data Mining Tools. *International journal of advanced research in computer science and software engineering*. vol. 4, no. 6. 2014.

[27] RapidMiner: *RapidMiner Studio Manual*. 2014. Retrieved from: https://docs.rapidminer.com/downloads/RapidMiner-v6-user-manual.pdf

[28] Relative Humidity. Online. Accessed on 2018-11-10. Retrieved from: http://hyperphysics.phy-astr.gsu.edu/hbase/Kinetic/relhum.html

[29] Relative Humidity in Air. Online. 2004. Accessed on 2018-11-11. Retrieved from: https://www.engineeringtoolbox.com/relative-humidity-air-d_687.html

[30] Rupp, R. F.; Vásquez, N. G.; Lamberts, R.: A review of human thermal comfort in the built environment. *Energy and Buildings*. vol. 105. 2015: pp. 178–205.

[31] Saeys, Y.; Inza, I.; Larrañaga, P.: A review of feature selection techniques in bioinformatics. *Bioinformatics*. vol. 23, no. 19. 2007: pp. 2507–2517.

[32] Saturation of air with water vapor. Online. Accessed on 2018-11-17. Retrieved from: http://www.atmo.arizona.edu/students/courselinks/fall16/atmo170a1s3/lecture_notes/humidity/saturation_figs/saturation.html

[33] Shapley, P.: Water in Atmosphere. Online. 2011. Accessed on 2018-10-17. Retrieved from: http://butane.chem.uiuc.edu/pshapley/GenChem1/L13/1.html

[34] Standard 55 - Thermal Environmental Conditions for Human Occupancy. Online. Accessed on 2018-11-01. Retrieved from: https://www.ashrae.org/technical-resources/bookstore/standard-55-thermal-environmental-conditions-for-human-occupancy

[35] Tang, J.; Alelyani, S.; Liu, H.: Feature selection for classification: A review. *Data classification: algorithms and applications*. 2014: page 37.

[36] The Heat Index Equation. May 2014. Accessed on 2019-02-08. Retrieved from: https://www.wpc.ncep.noaa.gov/html/heatindex_equation.shtml

[37] Trechsel, H. R.; Bomberg, M.: *Moisture control in buildings: the key factor in mold prevention*. chapter Moisture Sources. ASTM International. second edition. 2009. ISBN 978-0-8031-7004-9. pp. 103–109.

[38] Turris Gadgets. Online. Accessed on 2018-12-18. Retrieved from: https://doc.turris.cz/gadgets/

[39] Vafaie, H.; De Jong, K.: Genetic algorithms as a tool for feature selection in machine learning. In *Proceedings Fourth International Conference on Tools with Artificial Intelligence TAI'92*. IEEE. 1992. pp. 200–203.

[40] Vömel, H.: Saturation vapor pressure formulations. Online. December 2011. Accessed on 2018-11-10.
Retrieved from: https://www.eas.ualberta.ca/jdwilson/EAS372_13/Vomel_CIRES_satvpformulae.html

[41] What is Heat Index? Online. Accessed on 2018-11-05.
Retrieved from: https://www.weather.gov/ama/heatindex

[42] Zendulka, J.; Bartík, V.; Lukáš, R.; Rudolfová, I.: Získávání znalostí z databází ZZN. September 2009. Accessed on 2018-12-27.

[43] Zmrhal, V.: Ventilation, Psychometrics. Online. Accessed on 2018-11-29.
Retrieved from:
http://www.users.fs.cvut.cz/~zmrhavla/VENT/03_VEN_Psychrometrics.pdf

Appendix A

Contents of the Attached CD

Following directories and files can be found on the attached CD:

- directory `rm_processes` – RapidMiner processes used to design of detectors and predictor,
- directory `src_all` – a directory containing all the source code files,
- directory `src_Necasova` – a directory containing the created source code files,
- directory `text` – a directory containing the LaTeX source files of this thesis,
- file `README` – installation instructions,
- file `xnecas24.pdf` – an electronic version of this thesis in PDF format.