# Chapter 15
# Autonomic Source Selection for Real-time Predictive Analytics Using the Internet of Things and Open Data


**Ninad Arabekar, Wassim Derguech, Eanna Burke, and Edward Curry**

**Keywords** Source selection · Predictive analytics · Autonomic computing · Decision support · Internet of Things · Open data · Dataspaces

## 15.1 Introduction

Real-time predictive data analytics is a very important tool for effective decision support within intelligent systems. When making decisions using data, it is critical to use the most appropriate data. When creating predictive analytics, the selection of data sources is important as the quality of the sources influences the accuracy of the predictive model. Within a smart environment, a dataspace is valuable for data scientists as it provides a one-stop shop of all the data required for creating their analytical models: enterprise data, Internet of Things (IoT), sensor data, and open data. However, the increase in the number of data sources presents a challenge in selecting the most appropriate data source to use. The co-existence approach of dataspaces results in them containing much more data sources than within traditional data management approaches. This means that the need to perform source selection is an ongoing activity; as the dataspace is incrementally improved, sources will need to be re-examined to determine their suitability for tasks. We propose an autonomic source selection service for predictive analytics for intelligent systems within a smart environment. This service has been evaluated in real-world settings using a Real-time Linked Dataspace for energy predictions using IoT sensor data and open weather data.

The chapter is structured as follows: Discussion in Sect. 15.2 details the challenges of source selection for analytics. Section 15.3 provides an overview of the autonomic source selection approach, including its architecture and the suitability of prediction models. Section 15.4 details the source selection workflow and the criteria for reselection. In Sect. 15.5 we explore the source selection service together with machine learning models in two real-world intelligent systems. The chapter concludes with a summary in Sect. 15.6.



E. Curry, *Real-time Linked Dataspaces*,
https://doi.org/10.1007/978-3-030-29665-0_15

## 15.2 Source Selection for Analytics in Dataspaces

Real-time data sources are increasingly forming a significant portion of the data generated in smart environments. This in part is due to increased adoption of the Internet of Things (IoT) and the use of sensors for improved data collection and monitoring of daily activities in smart buildings, smart homes, smart cities, and others. In this section, we explore the need for new data support services to deal with the increased number of data sources and the resulting challenge of selecting the most appropriate real-time data source for building predictive models within intelligent systems.

### *15.2.1 Real-time Linked Dataspaces*

To support the interconnection of intelligent systems in the data ecosystem that surrounds a smart environment, there is a need to enable the sharing of data among intelligent systems. A data platform can provide a clear framework to support the sharing of data among a group of intelligent systems within a smart environment [1] (see Chap. 2). In this book, we advocate the use of the dataspace paradigm within the design of data platforms to enable data ecosystems for intelligent systems.

A dataspace is an emerging approach which recognises that in large-scale integration scenarios, involving thousands of data sources, it is difficult and expensive to obtain an upfront unifying schema across all sources [2]. Within dataspaces, data sources *co-exist* and are not necessarily fully integrated or homogeneous in their schematics and semantics. Instead, data is integrated on an *as-needed* basis with the labour-intensive aspects of data integration postponed until they are required. Dataspaces reduce the initial effort required to set up data integration by relying on automatic matching and mapping generation techniques. This results in a loosely integrated set of data sources. When tighter semantic integration is required, it can be achieved in an incremental *pay-as-you-go* fashion by detailed mappings among the required data sources.

We have created the Real-time Linked Dataspace (RLD) (see Chap. 4) as a data platform for intelligent systems within smart environments. The RLD combines the pay-as-you-go paradigm of dataspaces with linked data and real-time stream and event processing capabilities to support a large-scale distributed heterogeneous collection of streams, events, and data sources [4]. In this chapter, we focus on the source selection support service of the RLD. The selection of the correct data source is an important challenge in a dataspace. As the dataspace is incrementally improved, sources will need to be re-examined to determine their suitability for tasks. In Chap. 11, we explored the challenges of selecting event services based on their quality of service. In this chapter, we look at the classic source selection problem for creating predictive models from real-time stream analytics.

### 15.2.2 Internet of Things Source Selection Challenges

A multitude of Internet-connected devices generating data can quickly become infeasible to cope with. In a traditional data analytics scenario, decisions were driven by insights from information queried over statically stored tabular/relational data. However, with the increasing use of IoT devices, various business domains, governments (e.g. cities), and citizens can unlock the value of low-level data from sensor devices. Much of this data is now available as open data for public use. Choosing the right data source is an important part of effective decision-making within intelligent systems.

Optimal decisions can be made if only the most appropriate data streams are used within the decision-making process and predictive models [31]. However, it is seldom possible to manually decide in advance on the appropriate data sources for a specific application in a real-time big data streaming environment. Once decisions are made using data, it becomes crucial that the best quality data is used. Thus, it is imperative that data-driven decision models are built upon data streams that provide accurate and precise predictions while being tolerant of faults. Data quality issues in data-driven decision-making in critical domains can have disastrous consequences. The importance of source selection can be evident in intelligent systems which involve heavy presence of IoT sensors:

- *Autonomous Vehicles:* Semi/fully autonomous vehicles depend on IoT data streams for emergency roadside assistance, and real-time traffic alerts. The choice of right data streams can help these vehicles decide the best course of action. However, the selection of anomalous speed/direction/proximity sensors could result in accidents.
- *Wind Farm Energy Generation:* Multiple IoT sources from wind turbines, as well as open data sources for weather conditions and forecasts, need to be consulted to build efficient prediction models for wind farms [330]. However, defective sensors monitoring parts of power-generating turbines could lead to failure in maintaining the optimum performance.
- *Building Energy Management:* By leveraging the IoT sensors within a smart building, it is possible to predict the energy use of a smart building based on the weather forecast and the usage patterns of the building [25]. However, an error in the temperature monitoring system of a building could lead to wastage of fuel required for heating.

Another aspect that compounds the source selection dilemma for intelligent systems is the dynamic nature of data sources within an IoT-based smart environment. For example, given the task of real-time predictive modelling over high-velocity data streams, source selection is required to be quick, responsive, and autonomous in behaviour.

The problem of data source selection within a dataspace essentially boils down to identifying the most appropriate data streams that can be harnessed to build useful data models for descriptive as well as predictive analytics. The accuracy of these

predictive models forms the basis of selection criteria for the underlying data stream sources. Thus, a suitable approach is expected to be efficient and effective in the following aspects of the source selection problem:

- *Accuracy:* Use of machine learning models to achieve high accuracy.
- *Low Maintenance:* Source selection is autonomous, robust, and fault tolerant.
- *Highly Scalable:* Ability to withstand fluctuations in the number of data sources, data volume, or velocity.
- *Enrich Quality Metadata:* Update the quality of service of a data source, based on its performance for productive tasks.

## 15.3 Autonomic Source Selection Service for Real-time Predictive Analytics

The selection of data sources is important as the data from the sources influences the results of predictive analytics. In order to design our source selection service, we studied the available literature. In a 2011 review of trust in networked datasets [331], the authors noted that the process of selecting a data source is subjective based on the needs of the consumer. A conventional method for selecting a dataset to answer a query is to examine the metadata associated with the data source, for example, size of the dataset, date and frequency of updates [332]. Another method for determining correct information is to establish a consensus from several sources [331].

The co-existence approach of dataspaces results in them containing much more data sources than within traditional data management approaches. This means that the need to perform source selection is an ongoing activity; as the dataspace is incrementally improved, sources will need to be re-examined to determine their suitability for tasks. This constant change in dataspaces can be accompanied by rapid changes in data quality, which in turn affects their predictive power. Within the context of IoT, the scale of the data has increased, and for real-time predictive analytics, it is imperative that source selection should occur with minimum manual intervention. In order to meet these requirements, the source selection service of the RLD leverages techniques from autonomic computing to make the process as independent and self-managed as possible.

### *15.3.1 Autonomic Source Selection*

Autonomic computing systems are being developed to cope with large and increasingly complex systems. Autonomic systems can manage themselves when given high-level objectives from administrators by freeing them from low-level tasks [333]. The idea is to reduce the system operation and maintenance time to the minimum possible and allow the system to run at the best of its abilities. The four

**Table 15.1** Four pillars of an autonomic system [333]

| Trait | Explanation |
|---|---|
| Self-configuring | An autonomic application/system should be able to configure and reconfigure itself under varying and unpredictable conditions. |
| Self-optimising | An autonomic application/system should be able to detect suboptimal behaviours and optimise itself to improve its execution. |
| Self-healing | An autonomic application/system should be able to detect and recover from potential problems and continue to function smoothly. |
| Self-protecting | An autonomic application/system should be capable of detecting and protecting its resources from both internal and external attack and maintaining overall system security and integrity. |

pillars of an autonomic system (see Table 15.1) are self-configuration, self-optimisation, self-healing, and self-protection [334].

The selection service is designed to follow the principles of autonomic systems. The design of the selection service supports three of the four autonomic principles (self-protecting is not supported). The approach selects the data source by evaluating the results of the predictive analytics to determine the data source with the best results [335]. The approach maintains and improves the quality of the predictions over time while being self-managing [334]:

- *Self-configuration:*
  - *Automatic installation and initiation:* The source selection service can be installed into any prediction approach, and it automatically starts being useful with minimal intervention by a skilled worker.
  - *Generalisable:* There should be a low configuration effort to adapt the service to another prediction model to encourage re-use.
- *Self-optimisation:*
  - *Select the best data sources:* The service chooses the best sources of data to make the best possible predictions.
  - *Adapt to changes in the operation of the environments:* The predictions should react to changes in the operational phase, for example, expansion or contraction of a workforce or extensions/renovations to a building. Thus, source selection needs to adapt to operational changes.
  - *Low user interaction:* The service should continue working with no supervision.
- *Self-healing:*
  - *Transparent failover of a data source*: In the case of a failure of a data source (e.g. a weather station malfunction), the service should continue to make the best-effort prediction using an alternative data source so that agents dependent on it can continue to operate.

– *Maintain high-quality predictions*: Predictions must remain accurate as poor predictions may cause consumers of the data to make wrong decisions.
– *Timely identification of faults:* Faulty data sources (e.g. a damaged sensor) should be identified quickly so that an alternative data source can be used.

### 15.3.2 Architecture

The autonomic source selection service is designed according to the architecture depicted in Fig. 15.1. The service is part of the support services within the RLD, which is used to support the management of the data sources. The autonomic service is designed following the MAPE-K control loop from IBM [336] that consists of stages for Monitoring, Analyses, Planning, and Execution, all sharing a common Knowledge base.

Within the source selection service, these stages perform the following activities:

- *Monitor:* The monitor samples the outputs of the predictive models and stores the prediction for later comparison with the actual values. It is also responsible for

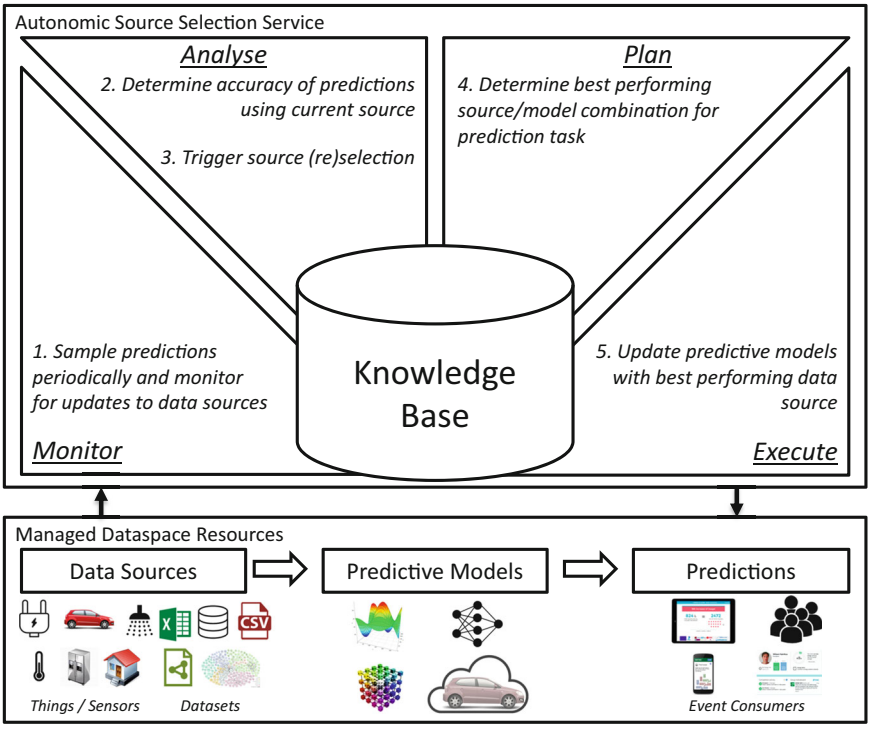


**Fig. 15.1** Autonomic source selection service following the MAPE-K control loop

observing any changes to the sources in the dataspace, such as new sources joining or updates to existing sources.
- *Analyse:* The main objective of this stage is to compare the predictions from the models with the actual readings to generate an error percentage, which is then used to determine the quality of the predictions. We use the Mean Absolute Percentage Error (MAPE) and Root Mean Squared Error (RMSE) as error indicators. The analysis is run at regular intervals to determine how well the models are performing.
- *Plan:* The planning stage is responsible for deciding when to select the sources used for prediction. Planning involves building many predictive models, which is costly and time-consuming, so trade-offs should be made between the frequency of reselections and maintaining the best prediction model possible. We suggest using hard limits of 15 min and 1 month for the upper and lower bounds of the reselect interval, but the specific interval should be kept dynamic. This planning activity builds a prediction model from the best available sources in the RLD.
- *Execute:* Updates the sources and prediction models that are using the source selection service within the dataspaces.
- *Knowledge Base:* The knowledge base is used to store and share data (e.g. source/model performance, and error rates) between the different stages in the MAPE-K.

### *15.3.3 Prediction Models*

We identified a set of requirements for choosing the right machine learning algorithm for the prediction models [335]. These requirements are listed below in descending order, from highest to lowest priority:

- *Accuracy:* The model should generate accurate predictions.
- *Fast Model Generation:* The model should be quickly generated.
- *Efficient with Minimal Data*: The model should be able to be deployed and quickly make accurate predictions. This requires the service to not overfit to the training set and result in drastically incorrect predictions.
- *Supports Nominal and Numeric Inputs*: Both nominal and numeric data will be used as inputs and need to be handled by the prediction model.
- *Generalisation Outside of the Available Training Data:* This is important as the prediction model will be used in a real-time scenario where data encountered will often be outside the range of data in the training set.
- *Low Configuration:* Minimal configurations effort required for a portable service.
- *Low Pre-processing of Data:* Pre-processing of the data does not require a skilled user. This is often automated when using a software suite.
- *Insights into Factors Influencing Prediction:* Dependency analysis is generated for user information and understanding.

**Table 15.2** Comparison of machine learning algorithms [335]

| | Multiple linear regression | Artificial neural network | Regression tree | Kernel regression analysis | Support vector machines |
|---|---|---|---|---|---|
| Insight into input importance | Yes | No (sensitivity analysis possible [337]) | Yes (tree shows which variables are important) | Yes | Yes |
| Overfitting prevention | Not prone to over fitting | Methods available | Pruning to stop overfitting may be required | Not prone to overfitting | Not prone to overfitting |
| Ease of implementation | Simple | Requires manual tuning of nodes and layers | Simple to understand and implement | Moderate | Moderate—optimisation exist |
| Computational cost | Low | Typically, high. Depends on training function | Low | Depends on training function | High on large data, scales $O(n^2)$ to $O(n^3)$ (adaptations available [338]) |
| Other benefits | Simple and quick | De facto solution for regression on non-linear data Extensive literature | Simple and quick | Works well outside of training data range | Works well outside of training data |
| Disadvantages | Poor with non-linear relationships | Prediction outside of training data can be drastically incorrect (corrections exist for this) Unimportant inputs may worsen predictions | Predictions not in continuous range-binned values | Needs normalising of input data | Needs normalising of input data |

Concerning these requirements, we carried out a comparison of the common machine learning algorithms found in the literature in order to understand their utility. The comparison is shown in Table 15.2.

The source selector needs to be scalable and efficient in its operation. The technical challenges this service faces are memory use, processing time, and latency. As the service generates multiple prediction models using training sets that potentially can span a considerable time, it is required to drop references to datasets as soon as they are not needed, so garbage collection to free memory can take place.

The processing time should be kept low by selecting sources with a greedy-type method. Evaluation of data sources is initially done over a large number of sources with small datasets, and it changes to a more comprehensive evaluation for fewer sources. As such, effort spent on poor data sources is reduced. Latency (from queries to a remote data store) is addressed by only querying the source for data that is necessary and by reducing duplicate queries where possible.

## 15.4 Autonomic Source Selection Workflow

The workflow for autonomic source selection service has two definitive stages: (1) initial model training with historical data, and (2) evaluation with real-time data streams. Both stages play an essential role in ensuring that the best data source from the dataspace is utilised for efficiently performing predictive model over real-time data streams.

**Stage 1: Initial Model Training with Historical Data** To forecast with real-time data streams, it is crucial that these models are tuned finely for different data sources as well as predictive algorithms are used. This stage aims at learning the optimum value of hyperparameters using various combinations of data streams and machine learning algorithms. The training happens with a significant number of historical observations accumulated over a considerable period. A large sample size helps in reducing bias (overfitting) while training the predictive models.

**Stage 2: Evaluation with Real-time Data Streams** Once the different predictive models are trained with optimum hyperparameters over multiple data streams, the same models are used for forecasting by using the real-time data feed. The models mostly try to predict the dependent values as close as possible to the observed ones. So, in the case of a classification problem, the F1 score would be the primary metric of evaluation. Also, specificity, recall, precision, or sensitivity might be considered depending on the type of classification task at hand. In the case of a regression task, the models are evaluated based on minimising the prediction error quantified with Root Mean Square Error (RMSE) and variance score. The data sources are then considered in increasing order of their RMSE scores (for regression) or decreasing order of F1 scores (for classification). The top-performing data source model is selected dynamically for predicting the outcome of real-time data streams.

### *15.4.1 4-Step Workflow*

Figure 15.2 is a step-wise representation of the autonomic source selection methodology. The essence of the approach is involving historical observations in learning about the quality of data source qualities and using machine learning to define their predictive power. These steps are elaborated below:

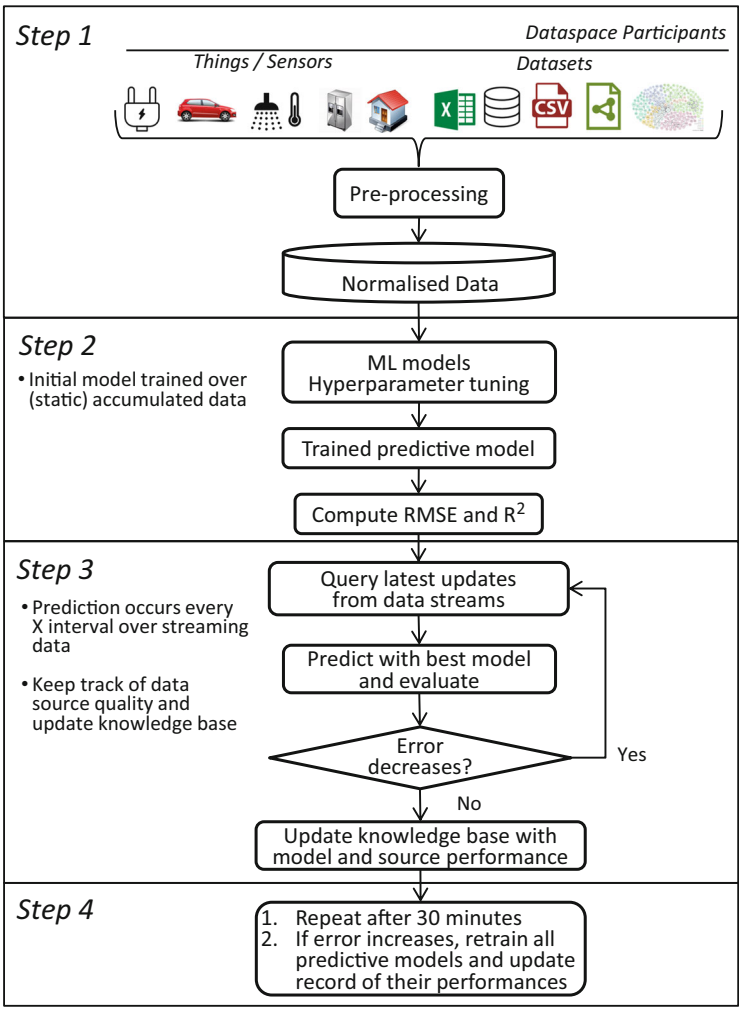


**Fig. 15.2** Workflow for autonomic source selection

- *Step 1—Autonomous Data Procurement and Pre-processing:* Attributes from the different sources (e.g. wind speed, wind direction, temperature) within the RLD are accumulated from different data streams for a period (e.g. 24 h). Since data sources and streams can serve data in different formats, an ETL (extract, load, transform) pipeline may need to be set up to collect and standardise it autonomously (see Chap. 6 for further details on normalising data sources in the RLD). The next part of the pre-processing is to aggregate and normalise the data. Data collected from multiple sources during the same period needs to be merged and used as features. Any data generated in the same duration from the prediction

target would need to be included here to serve as dependent variables explained by the features. This consolidated data would be aggregated on an interval basis to evaluate the predictive power of different sources during different times of the day. Since sources could return data reflecting similar conditions in the smart environment, collinearity between different features would be observed. This would help in detecting multi-collinearity issues in regression modelling of the prediction target. Also, relevant features (e.g. wind speed/direction/temperature) from different sources would be identified.

- *Step 2—Model Training (Hyperparameter tuning for ML models):* The aggregated data is now split in a 90:10 ratio with the 10% allocated for testing. The predictive models are trained, tested, and cross-validated to obtain different values of RMSE with an array of hyperparameter values for the Neural Network. Hyperparameters for which the lowest RMSE is observed after tenfold stratified cross-validation would be retained for predictive modelling with real-time streaming data.
- *Step 3—Model Evaluation (Iterative model evaluation with subsets of data sources):* RMSE and Mean Absolute Error (MAE) are calculated as a mean of relevant values obtained from cross-validation over the trained model. A subset of the data source with the least RMSE and MAE are retained, and others discarded. In subsequent iterations, other data sources are considered along with best performing one, to check if the error is minimised further. The adjusted coefficient of determination (Adj. $R^2$) which measures explained variance (while penalising the addition of new explanatory variables) would be taken into account. New data sources will be considered if RMSE and MAE are decreasing and adj. $R^2$ increases. The subset of data streams that minimise the error while maximising the explained variance is chosen as an optimum set of data sources initially. The performance evaluation of the models and data sources is stored in the knowledge base for future analysis.
- *Step 4—Dynamic Source (Re)selection:* It is possible that data streams return erroneous data or fail at a certain point in time. To counter such a situation, the service would build predictive models over a suitable time window for the prediction timeframe (e.g. 30 min for energy predictions) to check the RMSE and Adj. $R^2$ scores. In case the performance dips below a certain threshold, other data streams would be considered for building the model. If the performance metrics show improvement with new data sources, the original malfunctioning data stream would be replaced.

### *15.4.2 Reselection Triggers*

The criteria for the reselection of a source from the dataspace are shown in Table 15.3 and discussed further below.

**Table 15.3** Autonomic source reselection criteria [335]

| Reselection trigger | Reasoning | Mechanism | Autonomic characteristic |
|---|---|---|---|
| Timed error checks. | Checking if the prediction model has become less accurate. | Query the internal error knowledge base for the sources being used. | Self-optimising Self-healing. |
| Timed builds. (regardless of errors) | Data collected may allow a more accurate prediction model than the current one. | Send flag to reselect. | Self-optimising. |
| Very high error in the incoming stream. | May indicate a failure in a sensor or data source. | Short-term error check. | Self-healing. |
| New source event received. | New source may be more accurate than existing sources. | Send flag to reselect. | Self-configuring Self-optimising. |

**Timed Error Check** This is done by checking the recent performance against the expected performance of the prediction model. This check is for long-term trends in the dataset. Error checks are less costly than builds, so they happen on a more frequent basis than timed builds. The error check becomes more frequent every time the error returned is too high and less frequent when the error found is under the acceptable threshold. Changeable conditions make it check more often to include the newest and most relevant data and prevent data ageing. The error check is for 25% of the time the prediction model is in place so that the newest data is given high priority.

The error check uses Student's t-distribution to determine when the mean-error is too high. It does this by checking that the average error is lower than a threshold computed using: Threshold = (expectedMAPE) + 0.674 ∗ (standard deviation).

During the evaluation of source selection service, we found that this equation corresponds to a 75% one-tailed test, that is, 75% of predictions should be lower than this error %. The 0.674 figure is valid for more than 120 data instances, which corresponds to 30 h of data. For example, if the mean-error is 7.819% and the standard deviation of the error is 6.411%, the threshold would be: Threshold = (7.819) + 0.674∗(6.411) = 12.14.

**Timed Build** This is for the initial implementation phase of the service. It addresses the potential for improvement of the prediction model, whereas the error checking prevents degradation of the quality of predictions. The time interval begins at 15 min and increases by 50% every time a reselect flag is sent due to the time interval being exceeded. The time interval between reselections is reset if a reselect message is sent due to an error check or a new source detected.

**High Error Detected** It checks for deteriorations in the quality of the predictions such as a failure of a data source or any other error. This uses the previous formula, with 10% of the time the prediction model in place: Threshold = (expected MAPE) + 2∗(standard deviation).

**New Source Detected** This is activated if a listener picks up an observation from a new source. This adds the source to the pool of available options sooner than otherwise waiting for the reselect cycle, and it would be beneficial if a new dataset were added to the triple store in bulk with the addition of a new source.

## 15.5 Evaluation Within Intelligent Systems

The autonomic source selection service has been evaluated within two different real-world intelligent systems in the energy domain: (1) Wind Farm Energy Prediction, and (2) Building Energy Use Prediction. Both intelligent systems involved building predictive models using both IoT streams and managing open data within a Real-time Linked Dataspace.

### *15.5.1 Wind Farm Energy Prediction (Belgium)*

Wind power forecasting methods can be used to plan unit commitment, scheduling, and dispatch, and maximise profit by electricity traders [339]. We experiment with the utility of our source selection approach by selecting optimum data streams from multiple weather sources near the wind farms to predict the power generated. Predicting wind power while selecting the best from multiple weather data sources is an interesting challenge. The main set of features determining the wind energy generation is the weather conditions prevailing in the surrounding region. This useful information can be obtained from the data streams from weather stations in the vicinity. However, given the transient nature of this data, relying only on a single source can be potentially detrimental for consistently forecasting highly accurate power values.

The availability of real-time open streams on power generated from wind farms posed an initial problem. However, "Elia", one of the key electricity transmission system operators in Belgium, does a commendable job of publishing wind energy data frequently throughout the day via REST services as well as downloadable CSV extracts. We considered the weather updates released by the stations located in a 10 km radius range of this wind farm. As seen in Fig. 15.3, we have Elia-connected offshore wind farms (A) and four weather stations (B–E) located in the nearby coastal towns of Ostend, Zeebrugge, Middelkerke, and Knokke Heist. We performed a set of experiments with the source selection service choosing the best weather station to predict the output of the wind farm.

The results of experiments with different periodic prediction windows are summarised in Table 15.4. Although the trends in generated power seem to match perfectly with the wind speed recorded about 6 h ago, it is observed that the 1 h prediction window is the best indicator of power that would be generated from wind.


**Fig. 15.3** Locations of weather stations and wind farms

**Table 15.4** Source selection with periodic windows

| Metric | 1 Hour | 2 Hours | 4 Hours | 6 Hours |
|---|---|---|---|---|
| RMSE | 177.15 | 195.84 | 207.7 | 228.54 |
| Variance score | 0.66 | 0.57 | 0.52 | 0.44 |
| Correlation (Wind speed ~ power) | 0.74 | 0.72 | 0.66 | 0.60 |

Using the 1 h window size, multiple models are trained with combinations of relevant machine learning algorithms and weather data sources. Upon training the models, they are evaluated with RMSE and variance score metrics. The sources being transient are trained with algorithms that work well with regression tasks once the models are initially fitted with data; that is, the models fitted with an entire year of data are stored, and when a new weather observation is available on the data stream, they are used to predict the power that would be generated.

The best performing models are highlighted in Table 15.5. The performance is determined based on a low RMSE and a high variance score. The results achieved prove the effectiveness of autonomic source selection service. However, there are some limitations to the approach. For example, our experiments dealt with small volumes of low-latency weather streams. In this intelligent system, the serial training and testing process for the predictive models did not pose any significant performance issues. However, we envisage some performance degradation with high-velocity streams. Also, the approach relies on the data source being described in the catalog with the necessary metadata for their autonomous discovery. However, the source metadata may not pre-exist in some cases, and they would need to be created.

**Table 15.5** Summary of predictive model performance over 1-year data

| | Weather data sources (Weather stations locations) | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Zeebrugge (data source 1) | | Middelkerke (data source 2) | | Ostend (data source 3) | | Knokke (data source 4) | |
| Algorithm | RMSE | Var. | RMSE | Var. | RMSE | Var. | RMSE | Var. |
| Linear regression | 181.58 | 0.65 | **185.79** | **0.63** | 181.18 | 0.64 | 220.92 | 0.47 |
| Support vector machine | 183.58 | 0.64 | 196.17 | 0.59 | 186.09 | 0.62 | 196.87 | 0.58 |
| Artificial neural network | **158.57** | **0.73** | 202.94 | 0.56 | 168.39 | 0.69 | 223.73 | 0.46 |
| Decision tree | 182.72 | 0.64 | 206.45 | 0.54 | 175.83 | 0.66 | 210.19 | 0.52 |
| Ensemble (GBR) | 180.41 | 0.65 | 229.23 | 0.44 | **168.03** | **0.69** | **184.45** | **0.63** |
| Ensemble (AdaBoost) | 187.51 | 0.62 | 217.73 | 0.49 | 211.71 | 0.50 | 218.22 | 0.49 |

### *15.5.2 Building Energy Prediction (Galway, Ireland)*

The second intelligent system is at the Smart Building pilot at the Insight Centre at NUI Galway, Ireland. The building has been retrofitted with energy sensors to monitor the consumption of power within the building, including the consumption of devices, light, and heating. All the information from the building is managed within an RLD [100].

The first experiment investigates the accuracy of the service after the initial installation, that is, with no historical weather or electrical power data. Errors in the predictions versus the actual power readings are observed over time (errors for the 3, 6, and 12 hour-ahead predictions). Four machine learning algorithms in WEKA [340] were tested for short (1 week) and long-term datasets (5 weeks). The datasets were for the building's main incoming power and the weather observations of the NUI Galway weather station. For both datasets, the training set comprised of 66% of the available data and the remainder was used as the test set for evaluation. The datasets used for the testing contained the same attributes as the implemented model, that is, 15 min averages of power reading, time, the day of the week, temperature, pressure, humidity, wind speed, and wind direction. The datasets were randomised, and each experiment was performed three times, with average values taken. Unless stated, all configurations (see Table 15.6) are the defaults chosen by developers of the WEKA library ver. 3.7.3.

The results of the evaluation of the machine learning algorithm using short- and long-term datasets are shown in Tables 15.7 and 15.8, respectively. We notice from these tables that the Neural Networks, though more accurate, were far slower than the other two learning algorithms, and were not used in the service. The Linear Regression and Sequential Minimal Optimisation Regression (SMOReg) took approximately the same amount of time, with Linear Regression being more accurate. This may be due to the homogeneity of the summer dataset not presenting non-linear trends. For the implemented service, SMOReg was chosen due to the documented ability to handle non-linear data outside of the training set well, despite the inferior results from testing. The Artificial Neural Networks (ANNs) were

**Table 15.6** Algorithm configurations in WEKA for testing [335]

| Config number | Configuration settings |
|---|---|
| Config 1 | SMOReg. The WEKA implementation of a support vector machine for regression |
| Config 2 | One hidden layer backpropagation ANN with default WEKA values |
| Config 3 | Two hidden layer backpropagation ANN with default WEKA hidden layer one and ten nodes in hidden layer two |
| Config 4 | Linear regression. Default WEKA implementation for multiple linear regression |

**Table 15.7** Machine learning testing results for 1 week [Dataset size = 672, training data size = 443, test data size = 229] [335]

| | Mean absolute error (kW) | RMSE (kW) | Time (s) | Correlation coefficient |
|---|---|---|---|---|
| Config 1 (SMOReg) | 5.3638 | 6.9759 | 0.851 | 0.6233 |
| Config 2 (1 Layer ANN) | 2.7606 | 3.6242 | 47.004 | 0.9158 |
| Config 3 (2 Layer ANN) | 3.0473 | 4.1506 | 50.842 | 0.8961 |
| Config 4 (Linear Regression) | 4.8586 | 5.9283 | 0.759 | 0.7396 |

**Table 15.8** Machine learning testing results for 5 weeks [Dataset size = 3298, training data size = 2176, test data size = 1122] [335]

| | Mean absolute error (kW) | RMSE (kW) | Time (s) | Correlation coefficient |
|---|---|---|---|---|
| Config 1 (SMOReg) | 4.6755 | 6.5965 | 32.4 | 0.8054 |
| Config 2 (1 Layer ANN) | 3.3332 | 4.5841 | 229.7 | 0.9162 |
| Config 3 (2 Layer ANN) | 3.7566 | 4.7279 | 247.0 | 0.9221 |
| Config 4 (Linear Regression) | 4.7579 | 6.0173 | 2.4 | 0.8396 |

initially experimented on hourly power and weather readings for three months from October to December, where different combinations of hidden nodes were tested. During this testing, the second hidden layer with ten nodes was found to improve the RMSE by 1% over the single layer ANN. This improvement did not carry over to the more granular data in the real service, where adding the second hidden layer decreased the accuracy of the service.

## 15.6 Summary

In this chapter, we detail an autonomic source selection service for a dataspace to support the evaluation of real-time data streams for predictive analytics. The source selection service is designed using the principles of an autonomic system to reduce the administrative overhead. The service was developed and tested on two real-

world intelligent applications in the energy domain. The evaluation shows that the service was effective in supporting the choice of source and machine learning technique most appropriate to build predictive models in the energy domain.