Ontology-based decision tree model for prediction in a manufacturing network

Khan, Z. M. A., Saeidlou, S. and Saadat, M.

Ontology-based decision tree model for prediction in a manufacturing network

Zalan Mahmood Ayaz Khan, Salman Saeidlou & Mozafar Saadat

Taylor & Francis
Taylor & Francis Group

ARTICLE

Ontology-based decision tree model for prediction in a manufacturing network

Zalan Mahmood Ayaz Khan ⓘD, Salman Saeidlou and Mozafar Saadat

Department of Mechanical Engineering, School of Engineering, University of Birmingham, Birmingham, UK

ABSTRACT

This paper aims to create a predictive model, which will assist in the allocation of newly received orders in a manufacturing network. The manufacturing network, which is taken as a case study in this research, consists of more than 300 small manufacturing enterprises with a central company as the project managing integrator. The methodology presents the mapping of a PROSA (Product-Resource-Order-Staff Architecture) based ontology model on a decision tree, which was created with the Waikato Environment for Knowledge Analysis (WEKA) application. Furthermore, the methodology also demonstrates the formulation of the Semantic Web Rule Language (SWRL) rules from the WEKA decision tree with the help of MATLAB programming. The paper validated the result generated by the ontology model with the results of the decision tree model.

1. Introduction

In the late 1980s, when transportation became cheap, easier and fast, the manufacturing industry started to become globalized. Due to this globalization, the multinational companies connected their geographically dispersed plants by synergetic networks (Ferdows, 1989, 1997; Ghoshal & Bartlett, 1990). Apart from the multinational enterprise perspective, the ease of communication also enabled small enterprises to work with large enterprises. Large enterprises outsource their non-core competencies to small enterprises. Consequently, the collaboration of a large enterprise and a number of small enterprises make up a large manufacturing network (Jules, Saadat, & Saeidlou, 2013). A manufacturing network affiliates the manufacturers on an interim basis; they work together closely on a unique job and act together as a manufacturing service provider to a central company known as a project managing company (Jules, Saadat, & Saeidlou, 2015). A group of such manufacturing companies is known as a collaborative network organization (CNO) or a virtual breeding environment (VBE) (Camarinha-Matos, Afsarmanesh, Galeano, & Molina, 2009). The GFM (Gruppo Fabbricazione Meccanica) srl (GFM spa, 2018) is such an Italian-based project management company, who outsource their projects to more than 300 small and medium size manufacturing companies (Jules et al., 2013). Due to increase in the number of small and medium manufacturing companies, the GFM srl was facing the difficulty of scheduling the newly received orders

to the manufacturing companies. *The small and medium manufacturing companies are addressed as suppliers in this study.*

Therefore, this paper aims to develop an artificially intelligent model for GFM srl, which could predict the scheduling of a new order, which the company will receive in future, to an appropriate supplier. Appropriate suppliers are those who can complete the order and deliver it back in due time.

This paper is structured as follows: first, the literature review investigates a correct method for producing an intelligent predictive model, highlights the gaps in the research already done and formulates the objective of this study. Secondly, in the 'Methodology' section, a review of the computer applications used in this work will be presented. In the methodology, the three main steps followed by this research, namely data pre-processing, decision tree modelling and ontology modelling, will be described. The results and discussion will be based on the produced results before the conclusion is given.

2. Literature review

The idea of using a decision-tree-based ontology model in a predictive manufacturing system is a novel approach evidenced from literature. As there are a number of classification systems, like decision tree algorithms, etc. The decision tree algorithms are widely used in prediction and classification problems by several researchers because of their simple nature (Breiman, 2001; Chen & Guestrin, 2016; Cortez & Silva, 2008; Sambasivan & Das, 2017). In addition, research on the combined use of a decision tree model and ontology revealed that it is a trending topic amongst researchers in the field of classification and clustering (Jules et al., 2013; Mehta, Kshirsagar, Merchant, & Nair, 2015; Ravishankar & Shriram, 2013; Zhe Zhong, Saeidlou, Saadat, & Abukar, 2018). The reason for their use was that they found ontology-based clustering more efficient and flexible than a typical decision tree clustering.

In 2013, Guiovanni D. et al. used the same case study, which is used in this paper, for mapping the ontology model on the PROSA (Product Resource Order Staff Architecture). PROSA is a reference architecture for holonic manufacturing systems and was proposed in 1998. The PROSA introduced three main holons: product, resource and order. The staff holon can be used as an ad-hoc holon (Van Brussel, Wyns, Valckenaers, Bongaerts, & Peeters, 1998). The three basic holons in the PROSA architecture can be used to design any type of manufacturing system with all its important manufacturing functionality (Giret & Botti, 2009). In 2018, more developments were done by Zhong et al. on the case study of Jules et al. (2013). Zhe Zhong et al. (2018), which predicted the conformity of the orders with very high accuracy. They based their ontology model on PROSA architecture like Jules et al. (2013). But, their SWRL rules for logical reasoning were based on a decision tree. The decision tree was developed manually, which was only sufficient for small amount of data. On the other hand, Mehta et al. (2015) applied the concept of using decision-tree-based ontology to predict the completion of graduations in a university in Portugal. They used a decision tree to make a classification model for the prediction of graduation and used the same concept in making the ontology model. However, unlike Zhe Zhong et al. (2018) model, the decision tree in this work was created with WEKA, which is a powerful tool for developing a decision tree model.

In addition, the ontology model was created with Protégé, which is a popular tool for making ontology models. The SWRL rules were followed for the classification in the ontology.

Therefore, considering the gaps in the above-mentioned studies, in this work we have produced a predictive ontology model, which will be based on the decision tree algorithm, created with WEKA. It uses MATLAB for the automated extraction of the SWRL rules from the decision tree model. Unlike Zhe Zhong et al. (2018), the data will be imported to Protégé fully automatically with 'Cellfie plugin for Protégé'. The model is automated to a certain level, where minimum human intervention is required for either entering the data or creating the rules in Protégé.

2.1. WEKA workbench review

The Weka (Waikato Environment for Knowledge Analysis) workbench is an assemblage of up-to-date machine learning algorithms and data pre-processing tools, which was developed in the University of Waikato, New Zealand. It provides a number of data mining techniques like data preprocessing, feeding the data into the learning schemes, analyzing the classifier result, and able to visualize the result (Witten, Frank, Hall, & Pal, 2016). This paper uses WEKA as a widely employed preprocessing and classification tool for raw data. The paper uses WEKA's Gain Ratio Attribute Evaluator with Ranker as a search method for attribute selection and uses J4.8 as a classifier. J4.8 in WEKA is a slightly improved JAVA version of C4.5 revision 8 decision tree learner. The C4.5 is a divide-and-conquer algorithm, which is an improved version of the ID3 (Iterative Dichotomiser 3) decision tree algorithm. The ID3 uses information gain as an attribute evaluator and selects the root attributes on the bases of value numbers (Peng, Chen, & Zhou, 2009; J. Ross Quinlan, 1986). However, a new attribute selection was introduced in the C4.5 algorithm, called the gain ratio. The gain ratio takes the number of branches also into account before the split. (J Ross Quinlan, 1996, 2014; Witten et al., 2016)

2.2. Protégé review

Protégé is an open-source ontology editing software, and was developed by the Stanford Center for Biomedical Informatics Research at the Stanford University School of Medicine (Musen, 2015). Protégé has an attractive user interface, which enables users to make, edit and manage multiple ontologies. Furthermore, plugins can be added to Protégé, which makes it a fully customized framework (Noy et al., 2003). Some of the plugins which are used in this work are Pellet Reasoner, Cellfie and SWRL Tab.

3. Methodology

The flowchart in Figure 1 presents the methodology of the work done in this research. The elaboration of the main functions and steps of the flowchart are as follows:

3.1. Data pre-processing

Data from databases usually contain missing, noisy and conflicting information because sometimes they are generated by different sources or sometimes generated during a long span
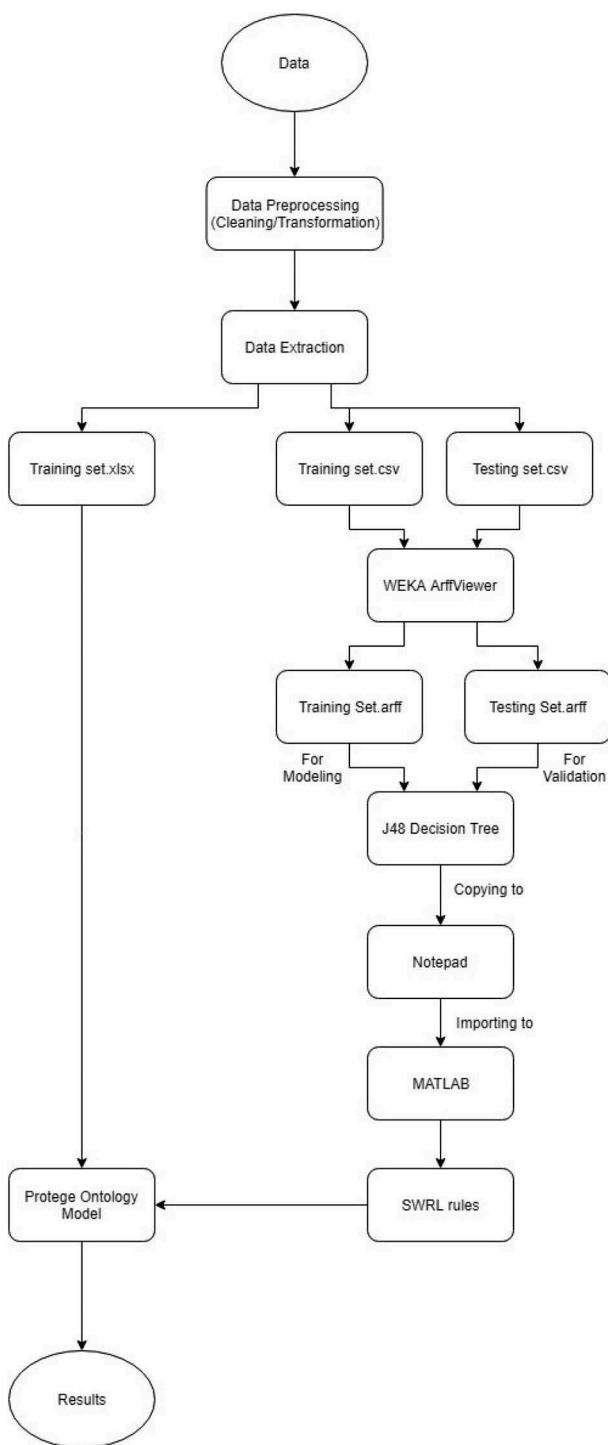
Figure 1. Flowchart demonstrating the methodology.

of time. Low quality of data will prompt low quality of data mining. Therefore, it is required to know the quality of data and clean it (Han, Pei, & Kamber, 2011).

3.1.1. Data cleaning and transformation

Figure 2 shows the raw data provided by GFM srl for this research, which was recorded from the years 2006 to 2016 and contained 102,219 orders completed by more than 300 suppliers. Each row represents a single order and the columns represent the information related to every order: such as, year and date the order was taken, supplier code, quantity (QTY), requested delivery date (RDD), actual delivery date (ADD).

Before importing the data to any software, the data were cleaned, for example the row with zero values was removed, changes were made to the values of the on-time

	A	B	C	D	E PO	F
1	PO_YEAR	PO_DATE	SUPPLIER CODE	PO NUMBER	POSITION	GFM CODE
2	2006	06/07/2006	2.3.4.10.370	2	10	2AE001031.00
3	2006	11/07/2006	2.3.4.10.370	3		2AEPT00001.00
4	2006	11/07/2006	2.3.4.10.370	3	10	6AE000199.00
5	2006	11/07/2006	2.3.4.10.370	3	20	6AE000199.00
6	2006	11/07/2006	2.3.4.10.370	3	30	6AE000199.00
7	2006	11/07/2006	2.3.4.10.7	4	20	2AE000385.00
8	2006	11/07/2006	2.3.4.10.7	4	40	2AE000380.00
9	2006	11/07/2006	2.3.4.10.258	5	30	2AE000367.00
10	2006	11/07/2006	2.3.4.10.453	6		CR00002016
11	2006	13/07/2006	2.3.4.10.113	7	10	2AE001145.00
12	2006	14/07/2006	2.3.4.10.31	8	10	GANSGCR037002
13	2006	14/07/2006	2.3.4.10.31	8	20	GANSGCR037002
14	2006	14/07/2006	2.3.4.10.31	8	50	GANSGCR037001
15	2006	14/07/2006	2.3.4.10.31	8	60	GANSGCR037001
16	2006	17/07/2006	2.3.4.10.370	9	820	2AE001108.00
17	2006	24/07/2006	2.3.4.10.7	10	10	2AE000849.00
18	2006	24/07/2006	2.3.4.10.7	10	20	2AE000848.00
19	2006	24/07/2006	2.3.4.10.7	10	30	2AE000827.00
20	2006	24/07/2006	2.3.4.10.7	10	40	2AE000850.00
21	2006	24/07/2006	2.3.4.10.7	10	50	2AE000783.00

G	H	I	J		K	L	M
QTY	REQUESTED_DELIVERY_DATE	ACTUAL DELIVERY DATE	LOT VALUE		DELAY DAYS	OTD	QUALITY
2	10/10/2006	04/12/2006	€	4,000.00	55	DELAY	CONFORM
1	10/07/2012	13/07/2012	€	-	3	DELAY	CONFORM
2	06/10/2006	06/11/2006	€	14,400.00	31	DELAY	CONFORM
2	06/11/2006	30/11/2006	€	14,400.00	24	DELAY	CONFORM
2	06/12/2006	29/01/2007	€	14,400.00	54	DELAY	CONFORM
8	24/07/2006	01/08/2006	€	38.80	8	DELAY	CONFORM
8	24/07/2006	26/07/2006	€	60.80	2	DELAY	CONFORM
2	24/07/2006	12/07/2006	€	24.00	-12	ON TIME	CONFORM
8	24/07/2006	26/07/2006	€	25.74	2	DELAY	CONFORM
2	17/07/2006	26/07/2006	€	120.00	9	DELAY	CONFORM
10	29/09/2006	17/10/2006	€	36,500.00	18	DELAY	CONFORM
10	24/11/2006	01/12/2006	€	36,500.00	7	DELAY	CONFORM
10	29/09/2006	17/10/2006	€	35,000.00	18	DELAY	CONFORM
10	24/11/2006	01/12/2006	€	35,000.00	7	DELAY	CONFORM
1	04/01/2007	08/11/2006	€	600.00	-57	ON TIME	CONFORM
44	06/11/2006	06/11/2006	€	568.04	0	DELAY	CONFORM
36	06/11/2006	06/11/2006	€	648.00	0	DELAY	CONFORM
40	06/11/2006	23/10/2006	€	516.40	-14	ON TIME	CONFORM
8	06/11/2006	13/11/2006	€	144.00	7	DELAY	CONFORM
4	06/11/2006	23/10/2006	€	51.64	-14	ON TIME	CONFORM

Figure 2. Original data (column A to M) from real industrial case study.

Table 1. Changes made to the OTD column.

RDDy	Delay days	OTD
<1 month	>0 days	Delayed otherwise On-Time
<3 months	>1 week	Delayed otherwise On-Time
<6 months	>2 weeks	Delayed otherwise On-Time
<1 year	>3 weeks	Delayed otherwise On-Time
>1 year	>1 month	Delayed otherwise On-Time

delivery (OTD) column and few extra columns were also added, such as requested delivery days (RDDy) and actual delivery days (ADDy).

In the original data, the OTD column was declaring all those orders as delayed, which added another day after the requested delivery date (RDD). Therefore, changes were made to the OTD column according to Table 1.

3.1.2. Data extraction

As mentioned above, the data contained more than a hundred thousand orders with more than 300 suppliers. For easier data manipulation and analysis, the model was trained and tested with only eight supplier's data. The combined number of orders for all the eight suppliers are about 21,700, which were made into two sets: a training and a testing set. The training set contained 66% of the total data, which were around 14,300 orders and the testing set contained the remaining 33% of the data, which were about 7,400 orders. The orders of each specific supplier were extracted from the original data with the help of MATLAB and exported to eight different Excel files. Then the eight files are combined into two Excel files namely, Training Set and Testing Set, in a ratio of 66% and 33%, respectively. Later, a copy of these two files was also made in the format of Comma-Separated Value (CSV). The Excel file was used by the Cellfie Plugin of Protege, and the CSV file was used to make an ARFF (Attribute-Relation File Format) file because the ARFF file is supported by WEKA.
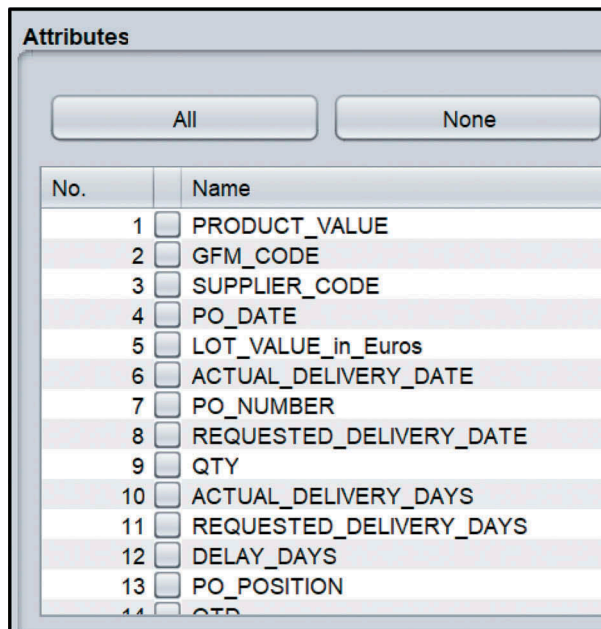
3.2. Decision tree

WEKA Software was used to make a predictive decision tree. Several steps were taken in developing the decision tree such as conversion of data to ARFF format, developing the decision tree with J4.8. These steps are explained below.

3.2.1. Conversion to ARFF format

Weka provides the ArffViewer tool, which imports the data from a CSV file and then can save it as an ARFF format. In this way, we converted the training set and testing set file, which were in CSV file format, into ARFF format.

3.2.2. Attribute selection

Later, the ARFF formatted training set file was fed to WEKA. The Gain Ratio Attribute Evaluator was used with Ranker as a search method. As the name itself defines, Gain Ratio Attribute Evaluator is an attribute evaluator used by the C4.5 algorithm. The Ranker search method, associate ranks to the attributes by their individual evaluation (Dinakaran & Thangaiah, 2013; Mehta et al., 2015; Witten et al., 2016). The evaluator sorted all the attributes on the bases of their importance, as shown in Figure 3.

Figure 3. Attributes sorted by the evaluator.

The attributes like GFM CODE, PO NUMBER, PO POSITION have nothing to do with our model. They were for the use and benefit of the company only. Therefore, these attributes were deleted from the list. Furthermore, the attributes like PO_YEAR and PO_DATE, requested delivery date (RDD), actual delivery date (ADD) represent the year and date on which the order was given, requested and delivered, respectively. These attributes cannot contribute to the prediction of a new order because a new order will have a new date; therefore, they were also deleted. The requested delivery days (RDDy) and actual delivery days (ADDy) are the derived attributes from the RDD and ADD, respectively, and they can be of importance to the model. However, the model was tried with the ADDy attribute in it, which gave an overfitted big tree. Over-fitting is a phenomenon in which the learning system tightly accommodate the given training data so much that it would unable to predict accurately the result of untrained data. Therefore, the ADDy attribute was also eliminated in order to avoid overfitting. The delay days and OTD (on-time delivery) attributes both tell whether the order was delayed or not, but the delay days was a numerical attribute and OTD was a binary attribute. Therefore, the delay days attribute was removed while the OTD attribute was left, as binary attributes are easy to classify (Han et al., 2011). Furthermore, the quality attribute was removed for two reasons: first, the ontology will predict the supplier name for a new order; therefore, new orders do not have conformity information with them; hence, they cannot be fed to the model. Second, we are already having an ontology model for the prediction of conformity of these data, which was in previous work done on this project and can be found here (Zhe Zhong et al., 2018). As a result, we were left with the only attributes shown in Figure 4, for creating a decision tree model. The supplier code was selected as a target attribute for the decision tree.
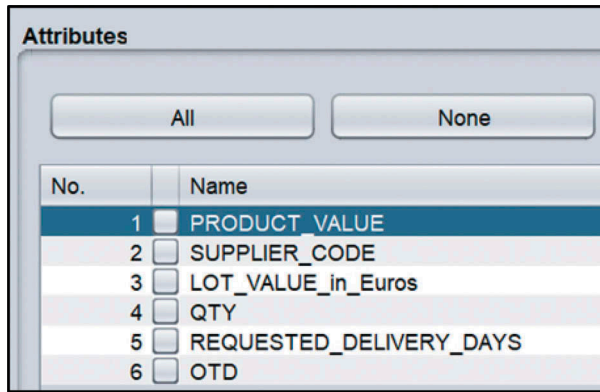
Figure 4. Final selected attributes.

3.2.3. Developing the decision tree with J4.8

With the selected attributes a decision tree was created with WEKA using J4.8 classifier algorithm. Despite the attribute selection algorithms already performed on the data set in the 'Attribute Selection' section, J4.8 will also apply its gain ratio attribute evaluator on the data. The training set was validated with the 10-fold cross validation. In 10-fold cross validation, the training set is divided into ten equal sets each having equal amount of target attribute. Nine out of the ten sets are used to train the model, and the last set is used as a testing set. This process continues for ten times until all the sets are used as testing sets; thus, it is called 10-fold cross validation (Witten et al., 2016).

The configuration of the J4.8 classifier was set as the following:

- confidence factor for pruning was set to 0.001
- minimum number of objects per leave (minNumObj) was set to 50
- unpruned option was set to false

They all are strategies used by WEKA to prune a decision tree. If the unpruned option was set to true, then the pruning of the tree will never take place. In addition, lowering the confidence factor decreases the post-pruning effect and eliminates the nodes which are not relevant. The minimum number of objects prevents the making of a new branch until the nodes in the branch are equal or greater than the specified threshold. Thus, this is a pre-pruning strategy (Drazin & Montag, 2012; Han et al., 2011; Rajput & Arora, 2013; Witten et al., 2016). Besides the above three options, all the rest of the options were left as default. The portion of the modelled decision tree is shown in Figure 6. This decision tree was used to create SWRL rules for the Ontology model.

Therefore, now it is time to jump to Protégé and model the Ontology part of the project and define SWRL rules for that model. The modelling and SWRL rules extraction are explained below in 'Generating SWRL Rules' section.

3.2.4. Ontology model

The ontology model was created in the Protégé software tool (Musen, 2015), which is based on the PROSA concept, and consists of five classes, namely, order, process, product,
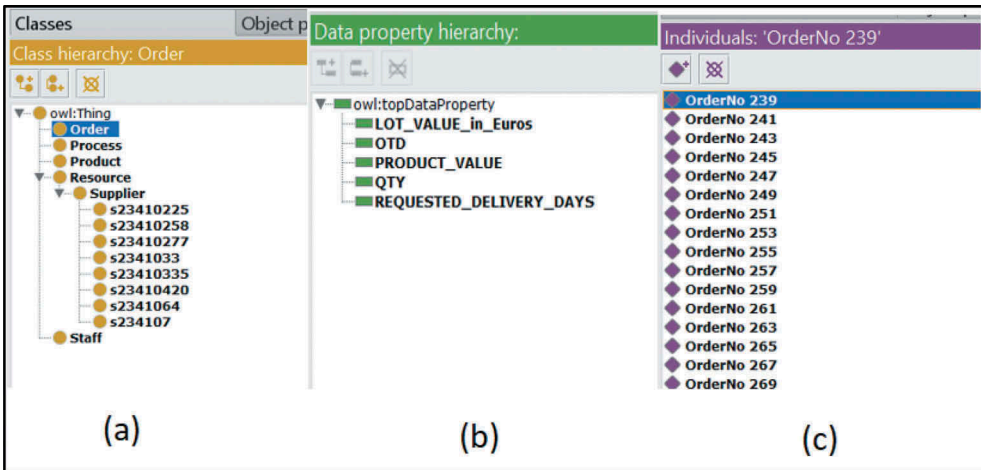
Figure 5. (a) Classes hierarchy (b) Data properties (c) Individual instances.
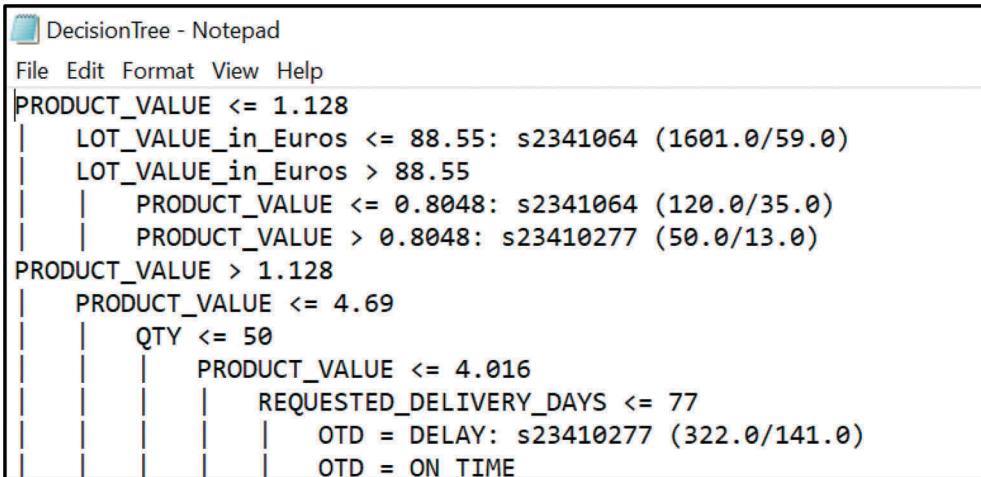
Figure 6. The decision tree mapped from WEKA to the notepad.

resource and staff. The suppliers were considered resources to the company. Therefore, the class 'Resource' has a subclass 'Supplier' and the 'Supplier' class further has the names of all the suppliers as subclasses. Figure 5(a) shows the classes in the Protégé environment.

3.2.5. Data properties

Figure 5(b) shows the data properties: LOT VALUE in Euros, OTD, PRODUCT VALUE, REQUESTED DELIVERY DAYS and QTY, which are same as the attributes in the WEKA model and headings of the columns in the original data.

3.2.6. Instances

Figure 5(c) shows the instances which were imported from the Excel file with the help of the Cellfie plugin of Protégé. The Cellfie plugin was used to import axioms from

spreadsheets to Protégé (Hardi, 2018a). Transformation rules were used to map the spreadsheet data to the ontology. The syntax used for the rules are Manchester OWL (Web Ontology Language) Syntax (Horridge & Patel-Schneider, 2009), which is a domain specific language (Hardi, 2018b).

3.2.7. Generating SWRL rules

After creating classes, data properties and instances in the ontology. We must define the SWRL rules for reasoning. The main challenge was to generate the SWRL rules from the decision tree model. Therefore, the SWRL rules were extracted from the decision tree in MATLAB. The decision tree which was built in the WEKA was mapped to a notepad, as shown in Figure 6, and a program was written in MATLAB which could read text from the notepad file and extract SWRL rules to another text file. MATLAB extracted each leaf of the tree as a single SWRL rule. For instance, the first rule in Figure 6 would be:

If PRODUCT VALUE ≤ 1.128 && LOT VALUE in Euros ≤ 88.55 THEN put the order in s2341064 (Supplier name).

The SWRL rule for the above rule would be:

Order(?od) ^ PRODUCT_VALUE(?od, ?P) ^ swrlb:lessThanOrEqual(?P, '1.128'^^xsd: decimal) ^ LOT_VALUE_in_Euros(?od, ?L) ^ swrlb:lessThanOrEqual(?L, '88.55'^^xsd: decimal) – > s2341064(?od)

The total number of rules generated for this model are 45, and the number are the same as the leaves of the decision tree. These rules can be copied to the SWRL tab plugin in Protégé. After generation of the SWRL rules, they are supposed to be run by the Pellet reasoner. The Pellet reasoner is an OWL-Description Logic (OWL-DL) reasoner with a sound reasoning efficiency and high popularity amongst its competitors. It is an open source and written in JAVA (Sirin, Parsia, Grau, Kalyanpur, & Katz, 2007).

4. Results and discussion

The ontology was based on the decision tree model, which was modelled in WEKA. It was therefore necessary to verify the ontology model against the given data, to check whether it would give the same exact result as the decision tree model.

The decision tree was modelled on the training set and was tested on the testing Set with the configuration of the J4.8 classifier, as shown in Table 2. In the first three trials, the confidence factor changes but the minNumObj is kept as default (i.e. 2). We can see how the number of leaves and size decreases by the converging of the confidence factor to a much smaller number, but the accuracy almost remained the same. However, lowering the confidence factor means we have less confidence in our training data (Rajput & Arora, 2013); therefore, the confidence factor was set fixed at 0.001. In addition, as mentioned in (Patel & Upadhyay, 2012), by increasing the minNumObj the size of the tree and number of leaves decreases dramatically with a very small amount of compromise on the accuracy, which can be seen in Table 3. Consequently, this configuration (i.e. confidence factor 0.001 and minNumObj 50) has been chosen as the final configuration. The details of this model are given in Figure 7.

The training set, with which the model was trained with 70.4% accuracy, contained 14,261 instances; and the testing set, with which the model was tested, contained 7,317 instances. Amongst the 7,317 instances, 61.6% (i.e. 4,508 instances) are classified correctly

Table 2. Different configuration of the J.48 classifier with its respective results.

Number of Trials	Confidence Factor	minNumObj	No. of Leaves	Size of Tree	Accuracy %
1	0.25	2	1095	2189	62.5
2	0.01	2	402	803	62.7
3	0.001	2	216	431	62.5
4	0.001	10	134	267	62.1
5	0.001	20	85	169	60.2
6	0.001	50	45	89	61.6

Table 3. Confusion matrix.

a	b	C	d	E	f	g	h	<-- Classified as
2062	9	25	0	9	677	29	0	a = s234107
267	84	1	11	3	145	0	0	b = s2341033
121	1	841	0	0	115	0	0	c = s2341064
0	0	0	11	13	0	0	0	d = s23410225
94	0	0	4	106	32	1	0	e = s23410258
897	11	50	9	7	1395	2	0	f = s23410277
101	0	4	7	0	93	9	0	g = s23410420
13	0	38	0	0	20	0	0	h = s23410335

```
Correctly Classified Instances          4508          61.6099 %

Incorrectly Classified Instances        2809          38.3901 %

Kappa statistic                              0.4375

Mean absolute error                          0.1225

Root mean squared error                      0.2597

Relative absolute error                   68.2052 %

Root relative squared error               86.623 %

Total Number of Instances                7317


Number of Leaves        :       45


Size of the Tree        :       89
```

Figure 7. Detailed result of the chosen configuration of the J4.8 classifier.

and 38.39% (i.e. 2,809) are classified incorrectly. The number of leaves is 45 and the size of the tree is 89. The confusion matrix of the model in Table 3 presents the correctly and incorrectly classified instances for the specific classes. The instances highlighted blue are correctly classified and these correctly classified instances are on the diagonal. In the first row of the confusion matrix, the 2,062 instances (highlighted as blue) are the correctly classified; the 9 instances (highlighted as green) belong to s234107, but are incorrectly

```
=== Detailed Accuracy By Class ===

                TP Rate   FP Rate   Precision   Recall   F-Measure   MCC     ROC Area   PRC Area   Class
                0.734     0.331     0.580       0.734    0.648       0.391   0.751      0.600      s234107
                0.164     0.003     0.800       0.164    0.273       0.346   0.804      0.325      s2341033
                0.780     0.019     0.877       0.780    0.826       0.799   0.918      0.833      s2341064
                0.458     0.004     0.262       0.458    0.333       0.344   0.687      0.122      s23410225
                0.447     0.005     0.768       0.447    0.565       0.576   0.897      0.524      s23410258
                0.588     0.219     0.563       0.588    0.575       0.366   0.753      0.571      s23410277
                0.042     0.005     0.220       0.042    0.071       0.085   0.810      0.106      s23410420
                0.000     0.000     ?           0.000    ?           ?       0.765      0.080      s23410335
Weighted Avg.   0.616     0.201     ?           0.616    ?           ?       0.786      0.582
```

Figure 8. Detail accuracy by class.

Figure 9. The correctly classified (a) and incorrectly classified (b) instances of the supplier s23410258.

classified as instances of the s2341033 supplier; the 25 instances (highlighted as red) belong to s234107, but are incorrectly classified as instances of the s2341064 supplier and so on. The TP Rate (True Positive Rate) in Figure 8 shows the percentage of correctly classified instances of each class. Therefore, s2341064 has 78% of correctly classified instances, which is the highest amongst all the other suppliers. On the other hand, s23410335 performed the worst with 0% correct instances.

For the verification and validation of the ontology model, the supplier s23410225 was chosen because only 24 instances belong to this supplier in the testing set file and it is easy to verify all the 24 instances. Hence, all these 24 instances are fed to the ontology model, which can be seen in Figure 5(c). After the running of the Pellet reasoner on the ontology, the instances were classified as expected. Figure 9(a) below shows the 11 correctly classified instances and Figure 9(b) shows the 13 incorrectly classified instances; they are classified as instances of the s23410258 supplier. Sometimes, when the classifier produces an incorrect

classification on a specific input, either the data has noise or may require additional information to achieve a good classification (Breiman, 2001). Therefore, the input data properties of these incorrectly classified instances matches the data properties of the s233410258 supplier; which is why the ontology model put them in the s23410258 supplier class. Hence, the accuracy of the model can be increased with a further addition of information regarding the supplier with which it is dealing.

5. Conclusion

The research aims to create an artificially intelligent predictive model for a manufacturing network, which can assist a project managing company in allocating a newly received order to its suppliers. This aim was achieved by developing an ontology model based on decision tree algorithm, created with the WEKA tool, which was mapped into ontology with the help of SWRL rules. The SWRL rules were extracted from the decision tree model with the help of a MATLAB program. The model gave 60.4% prediction accuracy using 8 suppliers, which is considered as a solid foundation for a model aimed at illustrating the idea of allocation of a newly received orders. The model benefits the manufacturing network industry by accelerating their planning and scheduling of tasks. In addition, a good level of automation is achieved in this work, which limits the large amount of manual data entry and enables the model to be used for large amount of data.

6. Future work

The model will be applied in an industrial case study as the future work when it is further optimized using more supplier instances. Moreover, the extraction of further attributes will be investigated which will hugely impact upon the overall prediction performance.

The import of SWRL rules from MATLAB to the Protégé was the only step used in this work where the data were manually entered. This can also be eliminated as the future work by programming this whole work in JAVA language using the APIs (application program interfaces) of WEKA and Protégé, which as result will also discard the use of the MATLAB.

Disclosure statement

No potential conflict of interest was reported by the authors.

ORCID

Zalan Mahmood Ayaz Khan http://orcid.org/0000-0003-3617-3918

References

Breiman, L. (2001). Random forests. *Machine Learning*, 45(1), 5–32.

Camarinha-Matos, L. M., Afsarmanesh, H., Galeano, N., & Molina, A. (2009). Collaborative networked organizations – Concepts and practice in manufacturing enterprises. *Computers & Industrial Engineering*, 57(1), 46–60.

Chen, T., & Guestrin, C. (2016). *Xgboost: A scalable tree boosting system*. Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining, San Francisco, CA.

Cortez, P., & Silva, A. M. G. (2008) *Using data mining to predict secondary school student performance*. Paper presented at the Proceedings of 5th Annual Future Business Technology Conference, Porto.

Dinakaran, S., & Thangaiah, P. R. J. (2013). Role of attribute selection in classification algorithms. *International Journal of Scientific & Engineering Research*, 4(6), 67–71.

Drazin, S., & Montag, M. (2012). *Decision tree analysis using weka. Machine learning-Project II* (pp. 1–3). Retrieved from http://www.samdrazin.com/classes/een548/project2report.pdf

Ferdows, K. (1989). Mapping international factory networks. *Managing International Manufacturing*, 3(p), 21.

Ferdows, K. (1997). Made in the world: The global spread of production. *Production and Operations Management*, 6(2), 102–109.

GFM spa. (2018). Retrieved from http://gfmspa.com/en/index.html.

Ghoshal, S., & Bartlett, C. A. (1990). The multinational corporation as an interorganizational network. *Academy of Management Review*, 15(4), 603–626.

Giret, A., & Botti, V. (2009). Engineering holonic manufacturing systems. *Computers in Industry*, 60(6), 428–440.

Han, J., Pei, J., & Kamber, M. (2011). *Data mining: Concepts and techniques.* Massachusetts, USA: Elsevier.

Hardi, J. (2018a). Cellfie plugin. Retrieved from https://github.com/protegeproject/cellfie-plugin/wiki.

Hardi, J. (2018b). MappingMasterDSL. Retrieved from https://github.com/protegeproject/mapping-master/wiki/MappingMasterDSL.

Horridge, M., & Patel-Schneider, P. F. (2009). OWL 2 web ontology language manchester syntax *W3C Working Group Note: Copyright © 2009 W3C® (MIT, ERCIM, Keio).*

Jules, G. D., Saadat, M., & Saeidlou, S. (2013). *Ontological extension of PROSA for manufacturing network formation.* Berlin, Heidelberg: Springer.

Jules, G. D., Saadat, M., & Saeidlou, S. (2015). Holonic ontology and interaction protocol for manufacturing network organization. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 45(5), 819–830.

Mehta, T., Kshirsagar, T., Merchant, A., & Nair, S. (2015). Graduate prediction using ontology approach. *International Journal of Computer Science and Information Technologies (IJCSIT)*, 6 (5), 4782–4784.

Musen, M. A. (2015). The protégé project: A look back and a look forward. *AI Matters*, 1(4), 4–12.

Noy, N. F., Crubézy, M., Fergerson, R. W., Knublauch, H., Tu, S. W., Vendetti, J., & Musen, M. A. (2003). *Protégé-2000: An open-source ontology-development and knowledge-acquisition environment.* AMIA … Annual Symposium proceedings. AMIA Symposium, Washington, DC.

Patel, N., & Upadhyay, S. (2012). Study of various decision tree pruning methods with their empirical comparison in WEKA. *International Journal of Computer Applications*, 60(12), 20–25.

Peng, W., Chen, J., & Zhou, H. (2009). *An implementation of ID3-decision tree learning algorithm.* Retrieved from web. arch. usyd. edu. au/wpeng/DecisionTree2.pdf

Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning*, 1(1), 81–106. doi:10.1007/BF00116251

Quinlan, J. R. (1996). *Bagging, boosting, and C4. 5* (Vol. 1). AAAI/IAAI.

Quinlan, J. R. (2014). *C4. 5: Programs for machine learning.* Elsevier.

Rajput, S., & Arora, A. (2013). Designing spam model-classification analysis using decision trees. *International Journal of Computer Applications*, 75(10), 6–12.

Ravishankar, T. N., & Shriram, R. (2013). *Ontology based clustering algorithm for information retrieval.* 2013 Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT), Tiruchengode, India.

Sambasivan, R., & Das, S. (2017). Big data classification using augmented decision trees. *arXiv preprint arXiv:1710.09567. Retrieved October 01, 2017*

Sirin, E., Parsia, B., Grau, B. C., Kalyanpur, A., & Katz, Y. (2007). Pellet: A practical owl-dl reasoner. *Web Semantics: Science, Services and Agents on the World Wide Web*, 5(2), 51–53.

Van Brussel, H., Wyns, J., Valckenaers, P., Bongaerts, L., & Peeters, P. (1998). Reference architecture for holonic manufacturing systems: PROSA. *Computers in Industry*, 37(3), 255–274.

Witten, I. H., Frank, E., Hall, M. A., & Pal, C. J. (2016). *Data mining: Practical machine learning tools and techniques.* Massachusetts, USA: Morgan Kaufmann.

Zhong, Z., Saeidlou, S., Saadat, M., & Abukar, A. (2018) *Ontology and rule-based reasoning for intelligent predictive manufacturing.* Advances in Manufacturing Technology XXXII: Proceedings of the 16th International Conference on Manufacturing Research, incorporating the 33rd National Conference on Manufacturing Research, September 11–13, 2018, University of Skövde, Sweden.