Towards an Ontological Framework for Integrating Domain Expert Knowledge with Random Forest Classification

Sadeer Beden
Department of Computer Science
Swansea University
Swansea, United Kingdom
sadeer.beden@swansea.ac.uk

Arnold Beckmann
Department of Computer Science
Swansea University
Swansea, United Kingdom
a.beckmann@swansea.ac.uk

***Abstract*—This paper proposes an ontological framework that combines semantic-based methodologies and data-driven random forests (RF) to enable the integration of domain expert knowledge with machine-learning models. To achieve this, the RF classification process is firstly deconstructed and converted into semantic-based rules, which are combined with external rules constructed from the knowledge of domain experts. The combined rule set is applied to an ontological reasoner for inference, producing two classifications: (1) from simulating the selected RF voting strategy, (2) from the knowledge-driven rules, where the latter is prioritised. A case study in the steel manufacturing domain is presented that uses the proposed framework for real-world predictive maintenance purposes. Results are validated and compared to typical machine-learning approaches.**

***Index Terms*—Ontology, Semantic Technologies, Reasoning, Random Forest, Industry 4.0, Steel**

I. Introduction

Semantic Technologies (ST) such as Ontologies and Knowledge Graphs have become a prominent way of capturing, modelling, and enriching knowledge within a specific domain. Within smart manufacturing, semantic modelling and ontology engineering have paved a way of constituting rich, machine-understandable vocabularies, offering extensive features including virtual data accessing and querying, as well as the ability to simulate the cognitive problem-solving behaviour of domain experts through rules and reasoning that enable new knowledge to be inferred [1].

On the other hand, Machine Learning (ML) techniques have been widely adopted in manufacturing to improve process operations and increase automatization with a variable degree of success [2]. As highlighted recently [3], challenges in this field include the dynamic operating environment and the need for context-aware information such as operational conditions in a production environment.

For these reasons, the development of hybrid approaches that combine ML and ST is regarded as a promising solution to integrate state-of-the-art black-box models within a context-rich, human-interpretable knowledge base. This paper introduces a framework that enables a RF classification process to be carried out using ontological methodologies in order to integrate domain knowledge as part of the process.

Steel Use-case: Cold rolling is the process of thickness reduction of steel material to produce thin sheets that are coiled. During this operation, physical rotating rolls exploit material deformation to achieve the diameter reduction necessary but get worn in the process. Because of this, the rolls are often refurbished to remove the worn surfaces. The Steel Cold Rolling Ontology (SCRO) [4] has been introduced to capture and model the concepts and processes of steel cold rolling which we use for our study. This work focuses on combining SCRO with a random forest classification that aims to predict the optimal maintenance schedule for refurbishment of the rolls.

II. Literature Review

In this section, we review literature that use a hybrid approach of combining machine learning models with ontologies.

Rajbhandari et al. [5] introduce an ontology to cover the lack of systematic methods of formalising domain knowledge for image object identification purposes. The authors create Semantic Web Rule Language (SWRL) rules from a combination of (1) generalised rules containing domain knowledge that were extracted from literature and domain experts; (2) localised rules that defined threshold values to calculate instances and their corresponding feature classes. Localised rules often require an adaptation to new threshold values, which are obtained from a RF classification, using the *inTrees* Framework. This framework offers some model interpretation to extract a reduced set of pruned rules from the RF. Once the new threshold values are calculated, the rules are executed to assign instances to their appropriate landslide classes. In comparison, the scope of our paper is to achieve the RF classification process using ontological reasoning, where the SWRL rules denote the paths in the RF.

Hastings et al. [6] focus on evaluating the applicability of using ML to automate the classification of chemical data

S. Beden was supported by the Engineering and Physical Sciences Research Council [grant number EP/T517537/1] and by Tata Steel.

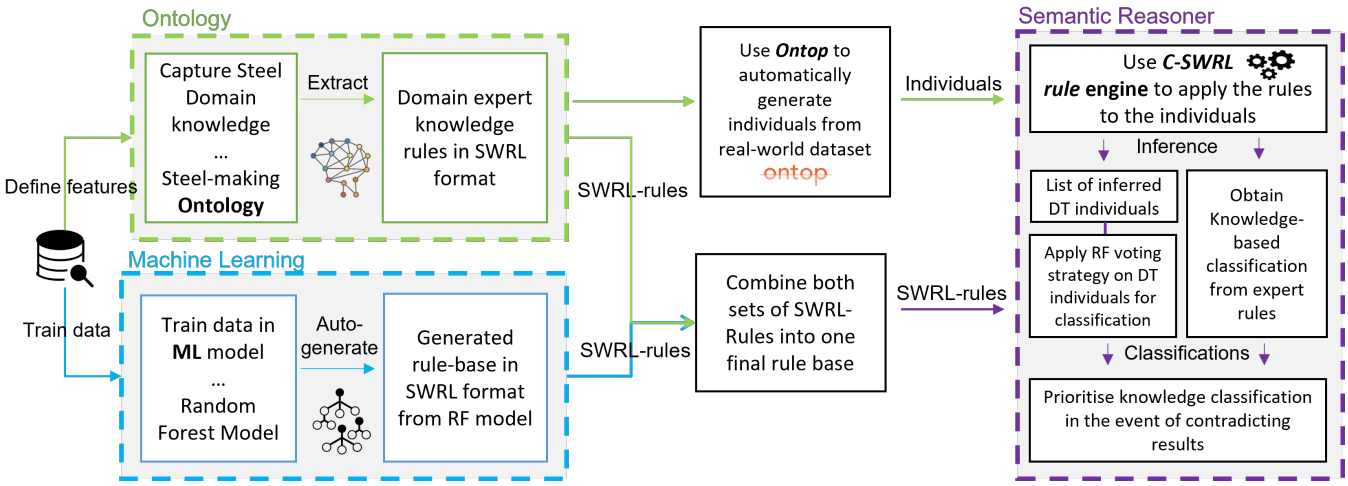

Fig. 1. Methodology of the proposed framework.

stored in chemical ontologies. The data is manually maintained so the authors survey and evaluate using different ML approaches to automate this process, giving a few examples including a RF implementation using the scikit-learn library. Other ML methods include: Logistic regression, K-nearest neighbour, DTs, Naive Bayes, linear discriminant analysis, and the support vector machine classifiers. For our work, our data is a combination of dynamic and static data that is automatically implemented into an ontological framework via Ontop Mappings.

Similarly, Johnson et al. [7] propose a data-driven rule learning method that follows a generic and interactive process that uses ontological-based knowledge to model qualitative knowledge onto a DT. This knowledge was obtained from domains experts and was inductively learned into the model itself using an iterative process until the domain experts were satisfied with the resulting DT. A case study of predicting food quality was demonstrated. For our work, we use an ensemble of DTs in the form of a RF instead of one DT. Our framework also aims to enable domain expert knowledge to be integrated directly into an RF without having to re-generate the DTs each time.

Finally, Cao et al. [8] combine both statistical and symbolic AI technologies to automate and facilitate predictive analysis within smart manufacturing. The paper introduces the Knowledge-based System for Predictive Maintenance in Industry 4.0 (KSPMI) that uses extracted chronicle patterns and machine learning techniques to infer machine degradation models from industrial data. Using KSPMI, SWRL rules are extracted from chronicle patterns that enabled ontology reasoning to automatically detect machine anomalies in smart manufacturing setting.

III. Methodology

There is much literature on semantic technologies and machine learning, but fewer when combining the two paradigms together. Naturally, ontologies play a prominent role in capturing and modelling domain knowledge, meanwhile machine learning discovers and exploits patterns in data without taking semantics into consideration. Figure 1 displays the flow of the proposed approach.

a) Machine Learning: Firstly, the data generated during the cold rolling processes are collected and imported into the RF model where a supervised learning method is used to train the data for classification. After the RF is constructed, each path in the RF is automatically converted into a unique if-then-rule in SWRL format, constructing a data-driven SWRL rule set.

b) Ontology: In parallel, the features selected in the RF model are captured and defined in the ontology with the help of domain experts, including their meta-data and relations.

Well-known knowledge acquisition methodologies such as interviews, questionnaires, task observation, protocol analysis, repertory grid analysis, concept mining, and many others [9] are adoptable in order to capture domain knowledge. In our example, we capture the attributes that contribute towards roll-wear during cold rolling using interviews and questionaires. This knowledge is stored in SWRL format, constructing knowledge-driven SWRL rules.

The data set is integrated into SCRO as virtual knowledge graphs using *Ontop* [10]. We are able to link each column in the database to a data property in the ontology using *Ontop Mappings* which generates an individual for each row.

c) Semantic Reasoner: In order to achieve the RF classification process in a semantic environment, it is necessary to use an ontological rule engine. During this process, the selected rule engine combines the data-driven and knowledge-driven rule sets into one. The individuals generated by Ontop are also passed through to the rule engine where the rule set is applied to each individual. Once the inference task is complete, two classifications will be produced: (1) from the RF voting strategy, (2) from the knowledge-driven rules. In the event of contradicting classifications, the knowledge-driven results are prioritised.

A. Data

To accurately predict the optimal maintenance schedule for the rolls, it is essential that all the impacting factors that contribute to roll wear are properly understood. To achieve this, several datasets have been collected from Tata Steel UK, and aggregated with the aim of linking the production information and chemical compositions of the steel coils to the reasons for changing the rolls. This new dataset, which we call *Roll_Unit_Trips_aggregated (RUTA)*, contains sensor information recorded with high resolution, production data including the chemical composition of the coils, historical refurbishment records, together with a tracking of the life-cycle of each roll.

B. ML Techniques

For the selected ML model, we have chosen *Random Forests* [11]. We implement our RF using the *scikit-learn* Python library [12] which adopts a soft voting technique to fully reproduce the RF classifier.

To build the RF classifier in our application, 75% of the original dataset (conformed by 3629 samples) was used as training set. The train-test split was performed randomly and in such a way that the original proportion of damaged rolls to not damaged rolls (20% - 80%) was respected. The value of the hyperparameters `n_estimators, max_depth` and `min_samples_leaf` was set using grid search and validating the *out of bag* score over the training set. The performance is as follows:

`Precision:`	0.83	`Weighted Precision:`	0.85
`Recall:`	0.41	`Weighted Recall:`	0.86
`F1-score:`	0.54	`Weighted F1-score:`	0.84

IV. Application

In order to simulate the RF classification process using ontology classification, the RF needs to be transformed into a set of SWRL rules. We have chosen the following approach as SWRL rules are well studied and recognised as demonstrated in the literature review. The RF generated is firstly stored in plain text as shown in Figure 2.

```
|--- trip_meterage <= 49.62
|   |--- cu <= 0.02
|   |   |--- weights: [0.00, 19.00] class: 1.0
|   |--- cu >  0.02
|   |   |--- weights: [3.00, 3.00] class: 0.0
|--- trip_meterage >  49.62
|   |--- cr_weighted <= 41.63
|   |   |--- weights: [9.00, 8.00] class: 0.0
|   |--- cr_weighted >  41.63
|   |   |--- stand_id <= 4.00
|   |   |   |--- weights: [5.00, 1.00] class: 0.0
|   |   |--- stand_id >  4.00
|   |   |   |--- weights: [12.00, 1.00] class: 0.0
```

Fig. 2. Structure of a scikit-learn RF

To read the RF, notice that each line contains a condition that includes one feature. If the condition is true, then continue to the next line, recursively. However, if the condition is false, then traverse down the pipe until the next condition.

Algorithm 1 SWRL-Rule Generation

```
I ← 0 {index of trees}
L ← [ ] {list of features}
for each tree in forest do
    for each node in tree do
        d ← depth of node in tree
        if node ≠ leaf node then
            L[d] ← node
        else
            R ← " " {string variable for forming a rule}
            for i = 0 to d do
                if i > 0 then
                    R += " ∧ "
                end if
                R += "L[i]"
            end for
            p ← probability {based on weightings}
            R += " → " + result(node, p, I)
        end if
    end for
    I += 1
end for
```

We have developed a basic algorithm that generates one SWRL rule for every path leading to a leaf node in the RF as shown in Algorithm 1. The antecedent of each rule contains all the nodes that satisfy that rule, including its features and conditions. Meanwhile, the consequent of the rule adopts *swrlx:_MakeOwlThing* from the SWRL-X library to generate a new *Decision_tree* individual that will store the prediction of the DT, along with the index of the tree in the RF. Below is an example of a SWRL rule of the first path from Figure 2.

Roll_Unit_Trip(?trip) ∧ hasTripMeterage(?trip, ?TripMeterage) ∧ swrlb:lessThanOrEqual(?TripMeterage, "49.62") ∧ hasCU(?trip, ?CU) ∧ swrlb:lessThanOrEqual(?CU, "0.02") ∧ swrlx:makeOWLThing(?DT, ?trip) -> Decision_Tree(?DT) ∧ isDecisionTreeOf(?DT, ?trip) ∧) hasPrediction(?DT, "1.0") ∧ hasTreeIndex(?DT, "1")

Then, as mentioned, the combined rule set that contains the data-driven rules and knowledge-driven rules are passed to an ontological rule engine for reasoning, producing the final classifications.

V. Use-cases and validation of our approach

A random forest R was implemented (see Section III-B) to illustrate the semantic framework introduced in the paper. R was constructed by a total of $K = 200$ DTs, containing over 34500 nodes, and averaging 58 leaf nodes per DT. These nodes contained a total of 120 different features that were used as predictors by the RF classifier.

We set up two use-cases for this experiment. The first use-case demonstrates the applicability of the proposed framework by deconstructing the RF into SWRL rules, then applying these rules to a semantic rule engine to examine if the classification results differ. No domain expert knowledge rules were added in this use-case. Meanwhile, we add simple domain expert rules as part of the second use-case.

In the first use-case, we simulated a scikit-learn RF classifier where each SWRL rule contained a probabilistic prediction between `0.0` and `1.0`, inclusively. Our application generated a

rule set containing a total of 11655 SWRL rules which denoted all possible paths in the RF. This rule set was not modified with any knowledge-driven rules. Then, when the rule engine was applied, a total of 200 `Decision_Tree` instances were generated, each of which contained a prediction. Afterwards, the mean probability of all 200 decision tree inferences was calculated to produce the final classification for each RUTA individual.

`Precision:`	0.93	`Weighted Precision:`	0.84
`Recall:`	0.45	`Weighted Recall:`	0.82
`F1-score:`	0.61	`Weighted F1-score:`	0.8

The statistics above are the results of performance in terms of precision, recall, f1-score, and weighted values of the proposed framework. The results are similar but not identical to the standard RF classifier mentioned in Section III-B. These results contained a slight divergence due to quantity of data used for validation. For the standard RF classification, the whole remaining 25% of the dataset was used for validation, whereas the semantic approach only used 101 rows from the dataset. With the experimental setup, we have noticed computational inefficiency and speed limitations of the approach. As our RF contained 200 DTs, each RUTA individual produced 200 intermediate DT individuals to simulate the RF classifier. As our study contained 101 individuals, this generated a total of 20200 DT individuals. Increasing the number of individuals beyond a certain threshold causes scalability and speed limitations due to the capabilities of the chosen reasoner.

Meanwhile, in the second use-case, we add domain expert knowledge to the framework. We have discovered that knowledge acquisition is a time consuming and strenuous topic itself. A great amount of knowledge that domain experts retain regarding cold rolling processes are considered of tacit nature, which require specific knowledge acquisition methods to extract. Presently, we are limited to testing the framework with simple knowledge-driven rules until more knowledge acquisition sessions are held. As an example, one simple knowledge-driven rule is *"If the selected steel grade is 5 and the total trip tonnage reaches beyond 6000, then the rolls will require refurbishing."* We translate this rule into SWRL format:

Roll_Unit_Trip(?trip) $\wedge$ *hasSteelGrade(?trip,?SteelGrade)* $\wedge$ *swrlb:Equal(?SteelGrade, "5")* $\wedge$ *hasTripTonnage(?trip, ?TripTonnage)* $\wedge$ *swrlb:GreaterThan(?TripTonnage, "6000")* -> *hasDamageFlagKnowledge(?trip, "1.0")*

We add all the knowledge driven rules into the rule set. After executing the rule engine and applying a soft voting strategy to the results, two classifications were produced for every individual: (1) *hasDamageFlag* from applying the voting strategy, (2) *hasDamageFlagKnowledge* from the knowledge-driven rules. In all occurrences of this use-case, the two results were identical as our knowledge driven rules were limited. Because of this, no validation is possible at this stage, but this use-case confirms that such a hybrid approach is possible and feasible.

VI. Conclusions and future work

This paper introduces an ontological framework that aims to enable the integration of knowledge-driven rules with RF classification. To achieve this, the RF classification is firstly deconstructed and converted into SWRL rules, before being combined with knowledge-driven rules extracted from domain experts. The final rule set is passed through an ontological reasoner for inference where two classifications were produced: (1) from simulating the selected RF voting strategy, (2) from the knowledge-driven rules, where the latter is prioritised in the occurrence of contradicting classifications.

Two use-cases were set up that deployed a scikit-learn RF classifier for predictive maintenance purposes. The first use-case demonstrated that the framework was successfully able to simulate the RF classification process using ontological methods such as SWRL-API. Some scalability issues were discovered and documented in the appropriate section.

Meanwhile, the second use-case displayed the ability to add simple domain expert knowledge with random forest classification. The framework highlights that such a hybrid approach is achievable and feasible but still require some final steps. The future work aims to bridge this gap by applying knowledge acquisition methods to extract more meaningful knowledge-driven rules from experts to apply to the framework. In parallel, we aim to develop a method to integrate the knowledge onto the data-driven rules directly by modifying the consequent of the rules, overcoming the black-box behaviour of RFs.

References

[1] Guohui Xiao and Linfang Ding. Virtual knowledge graphs: An overview of systems and use cases. *Data Intelligence*, 1:201–223, 05 2019.

[2] Jovani Dalzochio and Rafael Kunst. Machine learning and reasoning for predictive maintenance in industry 4.0: Current status and challenges. *Computers in Industry*, 123:103298, 2020.

[3] Yan Xu and Yanming Sun. A digital-twin-assisted fault diagnosis using deep transfer learning. *IEEE Access*, 7:19990–19999, 2019.

[4] Sadeer Beden, Qiushi Cao, and Arnold Beckmann. SCRO: A Domain Ontology for Describing Steel Cold Rolling Processes towards Industry 4.0. *Information*, 12:304, 07 2021.

[5] S. Rajbhandari and J. Aryal. Benchmarking the applicability of ontology in geographic object-based image analysis. *ISPRS International Journal of Geo-Information*, 6(12), 2017.

[6] J. Hastings and M Glauer. Learning chemistry: exploring the suitability of machine learning for the task of structure-based chemical ontology classification. *Journal of Cheminformatics*, 13(1), 2021.

[7] I. Johnson and J. Abécassis. Making ontology-based knowledge and decision trees interact: An approach to enrich knowledge and increase expert confidence in data-driven models. *Lecture Notes in Computer Science*, 6291 LNAI:304–316, 2010.

[8] Q. Cao and C. Zanni-Merk. KSPMI: A knowledge-based system for predictive maintenance in industry 4.0. *Robotics and Computer-Integrated Manufacturing*, 74, 2022.

[9] James R Heatherton and Todd T Vikan. An introduction to expert systems and knowledge acquisition techniques. Technical report, *Air Force Inst. of Tech Wright-Patterson AFB OH*, 1990.

[10] Guohui. Xiao. The virtual knowledge graph system Ontop. In *The Semantic Web – ISWC 2020*. Springer International Publishing, 2020.

[11] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[12] F. Pedregosa and Et al. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.