



Article

Steelmaking Predictive Analytics Based on Random Forest and Semantic Reasoning

Sadeer Beden ^{*}, Kayal Lakshmanan , Cinzia Giannetti  and Arnold Beckmann 

Faculty of Science & Engineering, Swansea University, Bay Campus, Swansea SA1 8EN, UK; k.lakshmanan@swansea.ac.uk (K.L.); c.giannetti@swansea.ac.uk (C.G.); a.beckmann@swansea.ac.uk (A.B.)
^{*} Correspondence: sadeer.beden@swansea.ac.uk

Featured Application: A framework to support steel operators in critical decision-making tasks.

Abstract: This paper proposes a human-in-the-loop framework that integrates machine learning models with semantic technologies to aid decision making in the domain of steelmaking. To achieve this, we convert a random forest (RF) into rules in a Semantic Web Rule Language (SWRL) format and represent real-world data as a knowledge graph in a Resource Description Framework (RDF) format, capturing the meta-data as part of the model. A rule engine is deployed that applies logical inference on the knowledge graph, resulting in a semantically enriched classification. This new classification is combined with external domain-expert knowledge to provide improved, knowledge-guided assistance for the human-in-the-loop system. A case study in the steel manufacturing domain is introduced, where this application is used for real-world predictive analytic purposes.

Keywords: steelmaking; ontology; knowledge graphs; semantic reasoning; machine learning; random forest; predictive analytics



Citation: Beden, S.; Lakshmanan, K.; Giannetti, C.; Beckmann, A. Steelmaking Predictive Analytics Based on Random Forest and Semantic Reasoning. *Appl. Sci.* **2023**, *13*, 12778. <https://doi.org/10.3390/app132312778>

Academic Editor: Paulo Santos

Received: 5 October 2023

Revised: 17 November 2023

Accepted: 19 November 2023

Published: 28 November 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Ontologies and knowledge graphs have become a well-established and recognised way of modelling and enriching knowledge within a particular domain. In the context of smart manufacturing, semantic technologies have become a promising solution for addressing Industry 4.0 challenges [1] and have many advantages including (1) the ability to provide a shared, machine-understandable vocabulary for data integration and exchange among components [2], (2) the capability to access and query data at a virtual level without physical data integration [3], and (3) simulating cognitive decision-making tasks through logical deductions, rules, and reasoning [4].

Meanwhile, machine learning (ML) models such as random forest (RF) have been widely adopted in manufacturing to optimize, control, troubleshoot, and improve process operations and automatization [5]. However, these models are faced with challenges such as the lack of context-aware information within dynamic production environments and semantic interoperability [6]. Thus, the development of hybrid models that combine semantic technologies and ML has been proposed to address these challenges.

Steel production is a particular example of a manufacturing process that requires extensive human knowledge, produces a vast amount of dynamical and static data, and where predictive analytics and maintenance are of utmost importance, usually associated with significant costs [7]. Cold rolling is one example of an important process in steelmaking, which is the process of reducing the thickness of steel strips to produce narrow sheets that are coiled. Work rolls are fundamental components of cold rolling, compressing the steel material, but they become worn after prolonged usage. They are required to be refurbished on a regular basis to remove the worn surfaces [8]. Presently, work rolls are refurbished based on the quantity of steel coils produced rather than their physical conditions. One

motivation of this study is to develop an application to aid operators on the shop floor in critical decision-making tasks in order to optimize the yield, efficiency, and overall life of work rolls.

In this paper, we demonstrate a hybrid model for predictive analytic purposes in the domain of steelmaking. The scheduling of the refurbishment process can then be targeted based on the condition of the rolls rather than an estimation based on the total tonnage produced. Additionally, anomalies and accidents within a steel plant such as spalling and overloads [8] can be identified and avoided pre-emptively with greater semantic interoperability.

To achieve this goal, we propose a human-in-the-loop framework that leverages knowledge representations and reasoning mechanisms using machine learning models to provide semantically enriched classifications, which are further combined with expert knowledge to support decision-making tasks. The support provided by the framework is similar to that of an expert system in the domain of steelmaking; however, the framework itself is not exactly an expert system and serves a more significant purpose.

The contributions of this paper are twofold: (1) we introduce the Random Forest Ontology (RFO) that captures and models a random forest at a conceptual level, which can represent and perform RF classification using rule-based reasoning and knowledge graphs, and (2) we demonstrate an iterative process to integrate external domain-expert knowledge with RF classification to provide more comprehensible decision-making assistance for the human-in-the-loop system.

The outline of this paper is as follows. Section 2 introduces the related works exploring existing hybrid models that combine semantic reasoning with ML. Section 3 introduces the methodology of our proposed approach. In Section 4, a use case is presented, where the framework is applied to assist steel operators, and the results of the application are utilized to validate the framework in Section 5. Finally, we end with the conclusions in Section 6 and future work in Section 7.

2. Related Work

There exists a significant amount of research that employs both semantic reasoning for inference tasks and ML for predictive tasks, but there are few works that combine the two paradigms. This section investigates the existing literature that combines ML models with semantic reasoning.

Rajbhandari et al. [9] introduced a hybrid model that combines ontology with random forest classification to address the lack of formalisation in systematic models for image object identification. Their model combines two sets of rules: (1) generalised domain knowledge rules from the literature and domain experts, and (2) localised rules obtained from an RF classification to classify landslides. In our paper, the RF classification is re-created using rule-based reasoning, where each rule denotes a path of an RF that is later combined with expert rules.

Similarly, Shoaip et al. [10] proposed an interpretable model to detect Alzheimer's disease using rule-based reasoning by combining the Alzheimer's Disease Diagnosis Ontology with a combination of different ML models. The Semantic Web Rule Language (SWRL) rules were obtained by combining a DT with the Java repeated incremental pruning model to produce a classification with enhanced reasoning efficiency. The rules were produced in a non-technical manner so that domain experts such as doctors could understand them and provide feedback without prior technical training. Meanwhile, in our framework, two distinct rule sets are employed. The first set comprises expert rules that represent the paths within the RF utilized for rule-based reasoning. The second set consists of domain-expert rules that are applied on top of any newly acquired classifications. Neither of these rule sets are displayed to domain experts.

Jabardi et al. [11] used ontological engineering and SWRL rules to identify and classify fake accounts on Twitter. The authors evaluated their ontology-based classifier results with different machine learning techniques, including naive bayes, logistic regression, and

Support Vector Machine (SVM), using the Waikato Environment for Knowledge Analysis (WEKA) tool. For this approach, the SWRL rules were manually written. The same authors expanded their research to cover DTs in [12] but did not cover how the rules were created. In contrast, the SWRL rules representing the RF in our paper are systematically generated through an algorithm introduced in Section 3, while our domain-expert rules are translated from natural language manually.

Johnson et al. [13] developed a method to model ontological-based knowledge into a DT through a generic and interactive process involving domain experts. Their method follows a data-driven rule-learning model that iteratively implements qualitative knowledge from the ontology into the DT until a complete DT is formulated. The authors exemplified the model through a case study focused on predicting food quality. Meanwhile, our framework integrates domain-expert knowledge with RF classification without the need to recreate the RF each time. Additionally, we follow an iterative process to access and validate our domain-expert rule set.

Sarkar et al. [14] introduced CHAIKMAT 4.0, a hybrid AI model that integrates semantic reasoning and machine learning paradigms to advance trusted flexible manufacturing aligned with Industry 4.0 goals. Their approach involves deploying deep learning and machine learning models for predicting machine capability and text analysis, and utilizes semantic reasoning to capture common-sense knowledge, enabling the generation of explanations for general tasks within a manufacturing production line. While the authors acknowledge existing technologies capable of achieving such goals, the implementation is left for future work.

Ammar et al. [15] introduced a proof-of-concept recommendation system that leverages machine learning and semantic technologies for explainability in AI. In their paper, the authors developed a hybrid prototype featuring a knowledge-driven recommendation system aimed at improving mental health surveillance based on adverse childhood experiences. The authors placed significant emphasis on ontological aspects and employed a question-answering agent from the Google DialogFlow engine to serve as a semantic knowledge base. The results in their prototype were compared to those of a ML classification, showcasing the added advantages of explainability. However, the hybrid method was still in the proof-of-concept phase and no concrete implementation was provided.

Bettini et al. [16] introduced proCAVIAR, a hybrid model that combines semi-supervised machine learning models with probabilistic knowledge-based reasoning for activity recognition. In their approach, the authors developed an ontology to capture the knowledge of various activities, including running, sitting, cycling and standing. Afterwards, probabilistic semantic reasoning was applied to comprehend these activities, and the outputs of the reasoner were combined with a ML classifier to generate a final prediction of the user's activity. In our work, we replicate the ML classifier itself using semantic reasoning.

Tofighi-Shirazi et al. [17] proposed a novel approach that combines semantic reasoning and ensemble machine learning classification for a framework designed to detect obfuscation transformations. The authors generated obfuscated samples and used semantic reasoning to extract raw data from these samples. The extracted data were then utilized to train various ensemble models for classification. In our study, we differ in approach, as we do not employ semantic reasoning to extract raw data. Instead, we utilize it to recreate the ensemble classification process of the RF using symbolic methods.

Pukkhem et al. [18] employed decision trees as the basis for generating an ontology with the objective of predicting the number of students graduating at a University. The DTs were created using C4.5/J48 algorithms. The authors emphasized how ontological representations play a key role for predictive purposes and the possibility of using SWRL rules to infer knowledge. However, the implementation details were not included and are unidentified as part of future work.

Finally, Cao et al. [19] integrated symbolic and statistical AI technologies for automation and predictive analysis within the domain of smart manufacturing. The authors adopt

ontological reasoning with statistical AI techniques using real-world datasets to generate a rule set in SWRL format. These rules were able to automatically detect machine anomalies within a shop floor. Meanwhile, our framework proposes different methodologies with results aimed at assisting operators on the shop floor.

3. Methods

Figure 1 displays the methodology of the proposed framework. There are three key components: Ontology, Machine Learning, and Semantic Reasoning which are described in the next sections.

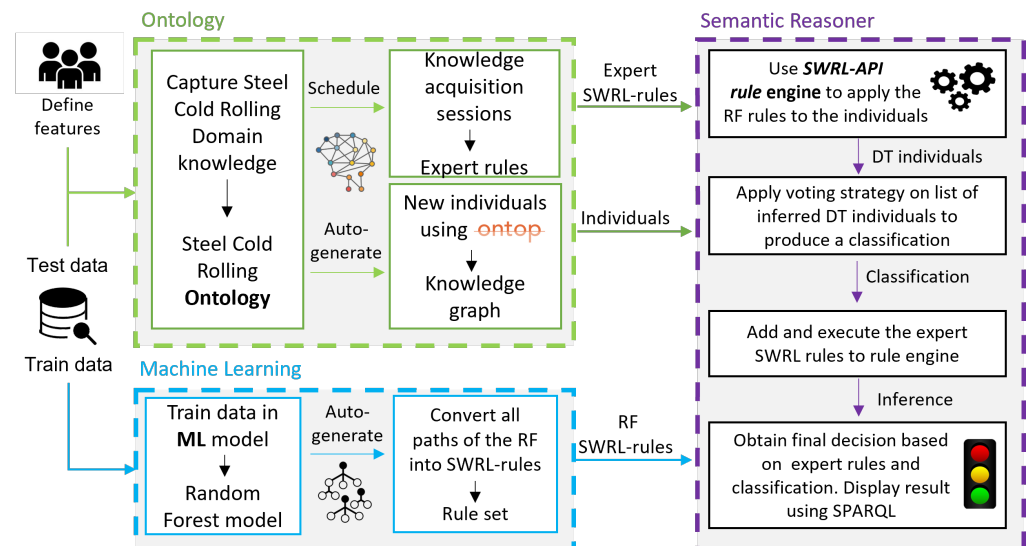


Figure 1. Methodology of our proposed framework.

3.1. Ontology

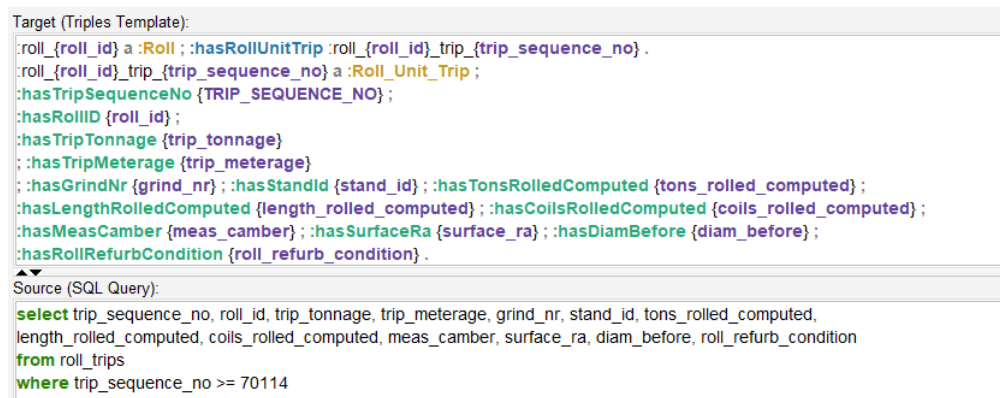
Ontologies have the capability to explicitly define concepts within a specific domain, along with their semantics [20]. Therefore, we leverage ontologies to capture the cold rolling processes and the semantics of the associated datasets. This work is built upon the Steel Cold Rolling Ontology (SCRO) introduced in [21]. SCRO models domain knowledge related to the cold rolling processes within a steel factory, emphasising semantic methodologies to store, access, and integrate real-world industrial data through virtual knowledge graphs. The cold rolling knowledge was acquired through interactions with domain experts and supplemented with relevant literature from online sources.

The initial step of the framework involves converting the dataset from a Structured Query Language (SQL) database into the Resource Description Framework (RDF) format to produce a knowledge graph. This knowledge graph is a collection of instances where the resources are identified, and their corresponding data values are represented as nodes. Meanwhile, their relations, or meta-data, are captured as edges, forming RDF tuples [22]. To achieve this, we employ the Ontop framework to automatically translate data into an RDF format [23]. Ontop offers two processes: (1) utilizing Bootstrappers that automatically convert an entire SQL table into an RDF format without explicitly defining any relations, or (2) using Ontop Mappings, which provide the functionality to select and filter specific data from an SQL table to correlate with existing properties in an ontology. In our framework, we chose Ontop mappings for added flexibility.

Ontop Mappings

There are two key components of an Ontop Mapping: (1) the Source containing an SQL query that allows the user retrieval of specific data through a `select` clause, (2) the Target which precisely maps the selected columns of a table to the chosen data or object property of an ontology.

Ontop Mappings are constructed manually and therefore require some knowledge of the syntax of the language. Figure 2 provides an example of an Ontop Mapping. In the source, we select which columns to include in the knowledge graph, while the target maps those features to the corresponding data properties in the ontology. The knowledge graph is then automatically generated by the materialization process in the Protégé IDE. The creation of the knowledge graph is the initial step in enabling semantic reasoning in our framework.



```

Target (Triples Template):
:roll_{roll_id} a :Roll ; :hasRollUnitTrip :roll_{roll_id}_trip_{trip_sequence_no} .
:roll_{roll_id}_trip_{trip_sequence_no} a :Roll_Unit_Trip ;
:hasTripSequenceNo {TRIP_SEQUENCE_NO} ;
:hasRollID {roll_id} ;
:hasTripTonnage {trip_tonnage} ;
:hasTripMeterage {trip_meterage} ;
:hasGrindNr {grind_nr} ; :hasStandId {stand_id} ; :hasTonsRolledComputed {tons_rolled_computed} ;
:hasLengthRolledComputed {length_rolled_computed} ; :hasCoilsRolledComputed {coils_rolled_computed} ;
:hasMeasCamber {meas_camber} ; :hasSurfaceRa {surface_ra} ; :hasDiamBefore {diam_before} ;
:hasRollRefrurbCondition {roll_refurb_condition} .

Source (SQL Query):
select trip_sequence_no, roll_id, trip_tonnage, trip_meterage, grind_nr, stand_id, tons_rolled_computed,
length_rolled_computed, coils_rolled_computed, meas_camber, surface_ra, diam_before, roll_refurb_condition
from roll_trips
where trip_sequence_no >= 70114

```

Figure 2. The mapping between SQL data and ontology using Ontop.

3.2. Machine Learning Models

Decision trees (DT) are capable of inferring rules from historical data to classify data-points as belonging to one of a predefined set of categories. They require relatively minimal data pre-processing steps and generate a definite set of simple rules that assign a unique category to each instance. During the training process, the DT progressively splits the dataset according to the value of a certain feature until a classification is reached for each resulting subset of the dataset. Thus, a DT can be viewed as a set of discrete rules. Each rule is composed of a logical conjunction of elementary formulae, accompanied by a class attribution. Any unseen instance satisfies precisely one of those rules, so that the classification performed by a DT is both human and machine interpretable. They are also advantageous as they have similar constructs to semantic web-based rules. However, DTs suffer from high variance and can quickly overfit the training set.

Meanwhile, a Random Forest (RF) [24] was introduced as a classifier to overcome these shortcomings. A RF classifier is formed by several DTs, each built using a randomly sampled subset of the training set, containing de-correlated predictors. The predicted category for each instance is determined by combining the output from the trees, which reduces the variance and hence improves the accuracy of the model. An algorithm known as a voting strategy is then used to calculate the final classification from the DTs. There are two main approaches: (1) using a majority voting strategy where the final classification is the modal value of all the DT predictions, or (2) using a soft voting strategy where the final classification is derived by calculating the average value of all the DT predictions.

We chose RFs over other ensemble methods due to their advantageous and easily comparable rule-like structure, which aligns well with the structure of semantic rules. This is particularly noticeable when contrasted with other machine learning models such as the Bayesian model. Random Forests have been extensively studied, and they demonstrated high accuracy across various complexities of classification and regression tasks [25]. Additionally, RFs provide a versatile set of techniques including bagging, node splitting, and various feature selection methods, further contributing to their suitability for our application [26].

The structure of a random forest is displayed in Figure 3, presenting a concise snippet in plain text that highlights the straightforward rule-like structure. Each line within the representation contains precisely one condition involving one specific feature. If the condition is met, the traversal continues to the next line, recursively. In case the condition is

not satisfied, the traversal instead travels down the pipe into a new line, typically involving the same feature with a reversed condition. This iterative process is repeated until a leaf node is reached, where weightings and classification information are stored.

```

|--- grind_nr > 53.50
|   |--- diam_before <= 573.40
|   |   |--- surface_ra <= 0.81
|   |   |   |--- weights: [0.00, 7.00, 1.00] class: 1.0
|   |   |   |--- surface_ra > 0.81
|   |   |   |--- weights: [102.00, 36.00, 66.00] class: 0.0
|   |--- diam_before > 573.40
|   |   |--- trip_tonnage <= 1832.38
|   |   |   |--- weights: [57.00, 35.00, 10.00] class: 0.0
|   |   |--- trip_tonnage > 1832.38
|   |   |   |--- weights: [30.00, 0.00, 5.00] class: 0.0

```

Figure 3. Structure of a scikit-learn RF in plain text format.

In this specific example, each leaf node is characterized by three potential weighted classifications enclosed in square brackets. The weighted values represent the quantity of training samples that satisfied all the conditions along that path up to the respective leaf node for each class. In a Breiman RF, the final classification is determined by selecting the maximum value among the three values, simulating a hard voting strategy. In contrast, a soft voting strategy would involve calculating the average of the three values to derive the final classification. This distinction in voting strategies adds to the flexibility and adaptability of the RF model.

Example 1. To illustrate the determination of a class, we consider the first leaf node in Figure 3. Such node is represented by the following information: *weights : [0,7,1] class 1.0*. This indicates that a total of 8 records from the training set satisfy that exact set of inequalities associated with this node. Among these, 0 belong to class 0, 7 belong to class 1, and 1 belong to class 2. Hence, in this example, class 1 is prioritized for both voting strategies, indicating its higher prevalence in the training set.

Within the framework, after training a RF for a classification task, the RF is stored in plain text format using the `export_tree` method provided by sci-kit learn [27]. Subsequently, an algorithm described below is utilized to convert the RF from plain text into semantic-based rules.

Random Forest Ontology and Algorithm

Random Forest Ontology (RFO) was developed to capture, model, and label the generic concepts of a random forest at a conceptual level. RFO includes fundamental classes of a random forest, such as `Random_Forest`, `Decision_Tree`, `Path`, `Voting_Strategy`, and others, displayed in Figure 4. RFO can be imported and combined with an existing ontology containing a knowledge graph to reproduce the RF classification process using semantic methods.

The first step of achieving RF classification based on ontological rule reasoning is by converting the RF into a format that supports logical inference and reasoning. We chose to adopt the Semantic Web Rule Language (SWRL) developed by the W3C consortium [28] to represent our rules as it is recognised as the leading rule language, which is well studied in the literature. In the framework, all paths of a RF are translated into SWRL rules by an algorithm we developed, displayed below as Algorithm 1, first introduced in [29]. The algorithm feeds in two lists as input, which creates a mapping between the features in the training set and their corresponding data properties in the ontology, e.g., `trip_tonnage` as `hasTripTonnage`. The algorithm then traverses through the random forest, creating a new rule for each path it finds and storing it in RFO.

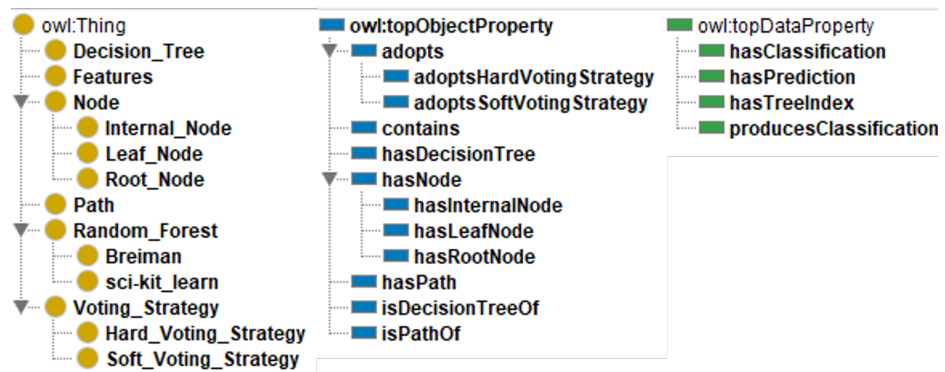


Figure 4. Classes, object properties and data properties of RFO.

Each SWRL rule contains the MakeOWLThing method from the SWRL-X library. When the rule is triggered, this method instantiates a new instance of the Decision_Tree class, and the resulting prediction is added to the knowledge graph. Additionally, a Random_Forest instance is generated and incorporated into the knowledge graph, establishing the connections between all the Decision_Trees in the RF, capturing their index.

Algorithm 1 SWRL-Rule Generation based on existing RF.

```

1:  $I \leftarrow 0$  {index of trees}
2:  $L \leftarrow []$  {list of features}
3: for each tree in forest do
4:   for each node in tree do
5:      $d \leftarrow$  depth of node in tree
6:     if node  $\neq$  leaf node then
7:        $L[d] \leftarrow$  node
8:     else
9:        $R \leftarrow ""$  {string variable for forming a rule}
10:      for  $i = 0$  to  $d$  do
11:        if  $i > 0$  then
12:           $R += "\wedge"$ 
13:        end if
14:         $R += "L[i]"$ 
15:      end for
16:       $p \leftarrow$  prediction {based on weightings}
17:       $R += "\rightarrow" + \text{result}(\text{node}, p, I)$ 
18:    end if
19:  end for
20:   $I += 1$ 
21: end for

```

3.3. Semantic Reasoning

Semantic Reasoners or Semantic Rule Engines are software that provide a mechanism for inferring logical deductions from a set of asserted axioms using a restricted set of first-order formulas [4,30]. In simple terms, a rule engine enables the creation of logical rules, which can be applied to a dataset to derive new knowledge from the existing knowledge [4].

The reasoning process involves two inputs: (1) an ontology containing a knowledge graph and (2) a rule set in an SWRL format. When the rule engine is executed, these rules are applied to the data entries in the knowledge graph, leading to the inference of new knowledge.

Thus, when the rule engine is inferred in the framework, each data entry yields N uniquely generated DT instances, where N corresponds to the size of the random forest. Afterwards, a voting strategy is applied for each data entry by calculating the average or modal value of the generated DT predictions. Algorithm 2 is an example of applying the

rule engine onto a Breiman RF containing three possible classifications, determining the modal value of these classifications.

Algorithm 2 Ontological Classifier for a Breiman RF with classes $0, \dots, n$.

Input: Ontology file incl. knowledge graph with individuals

Text file containing a list of SWRL rules generated from a RF

Output: Updated Ontology file with new acquired knowledge

```

1: load Ontology
2: let  $I1$  be a set of individuals relating to the relevant data instances
3: read in SWRL rules into rule engine
4: apply the rule engine
5: let  $I2$  be a set of individuals relating to Decision_Tree instances
6: for all  $r$  in  $I1$  do
7:   for all  $i$  from 0 to  $n$  do
8:      $C[i] \leftarrow 0$  {initialize count for class  $i$ }
9:   end for
10:  for all  $t$  in  $I2$  do
11:    if  $t$  is instance of Decision_Tree class for  $r$  then
12:      let  $p$  = prediction value of  $t$ 
13:       $C[p] += 1$ 
14:    end if
15:  end for
16:   $P \leftarrow \text{mode}(C[0] \& C[1] \& \dots \& C[n])$  {compute mode of highest class}
17:  store  $P$  as the hasClassification property for  $r$ 
18: end for
19: delete all intermediate rules
20: delete all intermediate Decision_Tree individuals
21: return new ontology file

```

Example 2. We consider a random forest with $n = 10$ decision trees. This can be explicitly represented in the RFO ontology with one instance of the RF class `RF_001` and n instances of the DT class, `DT_001` to `DT_n`. These DT instances are acquired by executing the rule engine on a rule set generated by Algorithm 1. Each instance of the DT class is linked to the RF individual via the `isDecisionTreeOf` relation, as well as its inverse relation `hasDecisionTree`. Furthermore, each instance of the DT class contains the data property `hasPrediction` to carry its classification. Afterwards, a voting strategy is executed to calculate the final classification of each individual by calculating the modal value or average of the `hasPrediction` values, as outlined in Algorithm 2.

Afterwards, domain expert knowledge can be integrated to produce a more comprehensible result that provides greater decision-making assistance for the human in the loop. To achieve this, the domain expert knowledge has to be acquired from domain experts using well-studied knowledge acquisition methods, and translated into an SWRL format so that logical reasoning can be applied. In essence, we are embedding the capabilities of a simplified expert system as part of the framework. These results are displayed using the SPARQL Protocol and the RDF Query Language [31].

4. An Application of the Framework: Use Case of Cold Rolling

This section demonstrates the applicability of the framework and introduces an example where the hybrid approach, utilizing real-world industrial data, is employed for predictive analytics purposes. Specifically, we present a use case illustrating how the framework can be applied in a cold rolling environment to assist operators in the crucial task of predicting the optimal time to stop operations to refurbish the rolls, providing helpful and actionable advice for the operator.

4.1. Application Use Case

As mentioned in the introduction, cold rolling of steel is one example of an important process in steel making: it is the process of reducing the thickness of steel to produce narrow sheets that are rolled into coils. During this process, the material undergoes deformation by passing through a set of rotating work rolls [32]. The work rolls are under constant pressure during operation, and they become worn after heavy usage, requiring regular refurbishment [8]. During the refurbishment process, the worn surface is removed, decreasing the diameter of the work rolls. The quantity of diameter removed, also referred to as stock, is calculated based on the quantity of steel produced rather than the physical condition of the roll. Because of this, the work rolls are often refurbished prematurely or belatedly; hence, their efficiency and yield are suboptimal. Therefore, it is important for operators on the shop floor to halt operations at an optimal point in time in order to maximize the yield of the work rolls, while simultaneously not overworking the work rolls, which in turn results in the production of defective steel and huge additional refurbishment costs. Typically, in our study, work rolls begin their life with a diameter of 600 mm and are scrapped when approaching 520 mm. Work rolls are expected to be refurbished hundreds of times before being scrapped, which varies between each work roll. An average refurbishment on a healthy work roll removes approximately 0.2 mm of stock. Meanwhile, if a work roll is damaged or over-worn, it may result in a significantly greater stock reduction. In extreme cases, an over 10 mm stock removal may be necessary, which is a significant cut in lifespan.

The acquisition of this knowledge involves active engagement with domain experts and stakeholders. Additionally, further interviews with domain experts are conducted to gather insights regarding the optimal timing to stop operation and refurbish the work rolls. Using this acquired knowledge, we construct a static set of expert rules designed to encapsulate these insights. This rule set is an essential component within the application, as these rules are applied to the real-time condition of the work rolls, influencing the final decision produced by the framework.

Thus, at any given point during cold rolling operations, the real-time conditions of the work rolls can be captured as a timestamp and input into the application. During this process, the data is integrated into a knowledge graph containing historical information about the work rolls, including details such as their previous grindings and stock reduction values. Subsequently, the knowledge graph is passed through a semantic reasoner, which applies logical deduction to predict the live condition of the work rolls. This prediction is accomplished through the application of a rule-based random forest classification, as introduced in this paper. Once this classification is obtained, it is combined with the expert rule set, generating a status for the operator and offering clear advice on whether to proceed with the cold rolling operations, along with insights into the recommended tonnage. Ultimately, the decisions produced by the application are intended to assist the human in the loop with their decision-making process and can be utilized as guidance.

In this use case, we apply the framework to the last 100 roll unit trips of our industrial partners and perform comparison with domain experts if the assistance provided is useful and accurate.

4.2. Producing a Random Forest

For this study, we deploy a supervised RF model to predict the condition of the work rolls at a given interval. The outcome can be one of three classifications: `class 0`, implying the condition of the work roll is *Bad*, i.e., the roll requires a considerable amount of stock reduction to remove the worn surface; `class 1`, where the condition of the work roll is considered as *Best*, and thus requires minimal stock removal; or `class 2`, implying the condition of the work roll is *Good* and an average stock reduction value is necessary.

There are many impacting factors that affect the rolls, which are collected and used for training and testing the RF model. This includes a combination of dynamic sensory and historical static data of the work rolls. The dynamic data contain live data read from sensors, which include the total tonnage and meterage rolled during a trip, speeds, temperature, as well as the coil usage data. This coil usage contains information regarding the steel grades and full chemical composition for each coil processed, e.g., its carbon or silicon values. Meanwhile the static data provide information regarding the roll historical data, such as their previous grindings, stock reductions, positioning, tons and length rolled, etc. This wide collection of data was explained by domain experts and data scientists from our industrial partners, which assisted in the data collection and aggregation aspects to build our random forest model.

To build the RF classifier in our application, 80% of the original dataset (9781 samples) was used as the training set. The train–test split was performed randomly and in such a way that the original proportion was respected. The value of the hyperparameters `n_estimators` and `max_depth` was set using grid search and validated through the performance of the metrics. Finally, the optimized values for `n_estimators` and `max_depth` were 20 and 22, respectively.

The RF contained a total of 20 decision trees that contained a total of 25,657 paths. The majority voting strategy technique was applied to calculate the final classification by computing the modal value of all the decision trees in the RF. The RF was exported to plain text using the `export_tree` method mentioned previously. The accuracy of the random forest is measured in terms of precision, recall and f1-score and their weighted values are displayed below.

Precision:	0.78	Weighted Precision:	0.78
Recall:	0.75	Weighted Recall:	0.78
F1-score:	0.76	Weighted F1-score:	0.77

When running Algorithm 1 for this particular RF, the 25,657 different paths produced an equal amount of SWRL-rules that were passed to the semantic reasoner for inference.

4.3. Reasoning

To build our knowledge graph, we created an Ontop Mapping that correlated the last 100 cold rolling trips from our local database into individuals in the ontology. These data entries were an instance of the `Roll_Unit_Trip` class.

First, the 25,657 rules and the 100 data entries were input into the reasoner, initiating the application of rules to the knowledge graph for logical inference. Listing 1 displays the syntax and format of one of the 25,657 rules as an example. As shown, the antecedent of every rule starts with `Roll_Unit_Trip(?trip)` to target the corresponding instances of the `Roll_Unit_Trip` class in the knowledge graph that the rule is applied to. Each instance must be linked to an instance of the RF class via the `RF0:hasRandomForest` property provided by RFO (which is typically mapped automatically). Then, each rule contains the features and conditions of the path, in this case `hasGrindNr`, `hasDiamBefore` and `hasSurfaceRA` and their conditions, respectively. The end of the antecedent exploits the `makeOWLThing` method from the SWRL-X library to instantiate a new, unspecified instance in the knowledge graph. Meanwhile, the consequent of the rule specifies the new instance to be of type `RF0:Decision_Tree`, which contains the `RF0:hasPrediction` data property to store the classification, as well as the `RF0:treeIndex` data property to store the index of the tree in the RF.

Listing 1. Example of an SWRL rule.

```

Roll_Unit_Trip(?trip) ^ RFO:hasRandomForest(?trip, ?rf) ^
hasGrindNr(?trip, ?GrindNr) ^ swrlb:greaterThan(?GrindNr, 53.50) ^
hasDiamBefore(?trip, ?DiamBefore) ^ swrlb:lessThanOrEqual(?DiamBefore, 573.40) ^
hasSurfaceRA(?trip, ?SurfaceRA) ^ swrlb:lessThanOrEqual(?SurfaceRA, 0.81) ^
swrlx:makeOWLThing(?DT, ?trip) -> RFO:Decision_Tree(?DT) ^ RFO:isDecision
TreeOf(?DT, ?rf) ^ RFO:hasPrediction(?DT, 1) ^ RFO:hasTreeIndex(?DT, 1)

```

After the rule engine completed its process, each `Roll_Unit_Trip` individual accumulated a total of 20 unique instances of the `RFO:Decision_Tree` class, each containing a prediction. Afterwards, as this RF utilised a majority voting strategy, the modal value of the 20 classifications was computed to derive the final classification for each `Roll_Unit_Trip` individual. This final classification was then stored in the ontology. Finally, all intermediate values, including the decision tree instances, were purged from the ontology, resulting in a streamlined and concise representation of the final classifications.

4.4. Limitations and Validation

The proposed method is computationally expensive for large RFs or large datasets. In our study, the sci-kit learn model was using the remaining 20% of the dataset (1957 samples) for validation, which, when converted into rule-based reasoning, was too large for the default SWRL-API reasoner `Drools`. Therefore, we instead compared the accuracies of a batch of 100 data points iteratively, which overall produced identical results for all data points to the sci-kit learn validation, validating our approach. More concretely, a total of 25,657 paths in the RF translated to 25,657 SWRL rules that were passed to a rule engine. Each rule was applied to the batch of 100 instances, producing a total of 2,565,700 inferences. Meanwhile, the authors in [33] compared the performance of different rule engines and concluded that `Drools` is optimised for smaller datasets and has the worst performance with larger datasets when compared to other reasoners. A possible solution for this performance issue is to investigate the use of a different rule engine that is compatible with SWRL-API.

4.5. Integrating Domain Expert Knowledge

As mentioned, numerous industrial processes within the steel domain heavily rely on knowledge, where plant operators constantly make important decisions based on the scenario and their expertise. Meanwhile, the framework combines domain expert knowledge with ML classification to offer decision-making assistance for the human in the loop. The initial step involves acquiring expert knowledge through one or more well-known knowledge acquisition methods. This knowledge must be captured in a format that is translatable into an SWRL format for compatibility with the rule engine. These expert rules are treated as highly accurate within rule-based systems [34].

Knowledge Acquisition Methods

Domain expert knowledge is often considered to be of implicit and tacit nature which contradicts ontologies explicit modelling behaviour [35]. In the context of manufacturing, tacit knowledge refers to the concept of informal learning by simply performing actions and experiences, often where the knowledge is unconsciously retained in individual memory rather than being formally recorded or shared [35]. To overcome this phenomenon, many different ways of extracting tacit knowledge have been widely studied over the years. This includes techniques such as interviewing, questionnaires, protocol analysis, inferential flow analysis, and many more [36].

For our study, we conducted interviews and questionnaires with domain experts and plant operators to construct our domain expert knowledge rule set. Presently, we obtained a small sample of domain expert rules which we aim to increase in quantity and quality over time. Meanwhile, the small sample of domain expert rules demonstrate the capabilities of the framework. Table 1 displays some expert rules in a categorised format before they are translated into the SWRL format. Example 3 displays a domain expert knowledge obtained

during the knowledge acquisition sessions which we translate as rules. As these rules come directly from knowledge acquisition sessions with domain experts, the validity and accuracy of these rules are treated as absolute. In an event that a rule creates an incorrect status, it is reviewed with experts and updated accordingly.

Table 1. Domain expert knowledge rules in a simplified view.

ID	Classification	Recent Severe Grinding	Tonnage Limit	Tonnage Reached	Status
1	Best	False	4500	False	Continue rolling: predicted condition is best on a healthy roll. Recommended tonnage of 4500.
2	Best	False	4500	True	Stop rolling: predicted condition is best on a healthy roll. Recommended tonnage of 4500 is exceeded.
3	Best	True	4000	False	Continue rolling: predicted condition is best but recent high stock removal may affect roll. Recommended tonnage of 4000.
4	Best	True	4000	True	Stop rolling: predicted condition is best but recent high stock removal may affect roll. Recommended tonnage of 4000 is exceeded.
5	Good	False	4000	False	Continue rolling: predicted condition is good on a healthy roll. Recommended tonnage of 4000.
6	Good	False	4000	True	Stop rolling: predicted condition is good on a healthy roll. Recommended tonnage of 4000 is exceeded.
7	Good	True	3500	False	Continue rolling: predicted condition is good but recent high stock removal may affect roll. Recommended tonnage of 3500.
8	Good	True	3500	True	Stop rolling: predicted condition is good but recent high stock removal may affect roll. Recommended tonnage of 3500 is exceeded.
9	Bad	False	3000	False	Continue rolling: predicted condition is bad on a previously healthy roll. Recommended tonnage of 3000.
10	Bad	False	3000	True	Stop rolling: predicted condition is bad on a previously healthy roll. Recommended tonnage of 3000 is exceeded.
11	Bad	True	2500	False	Stop rolling: predicted condition is bad on a roll that was previously damaged.
12	Bad	True	2500	True	Stop rolling: predicted condition is bad on a roll that was previously damaged. Recommended tonnage of 2500 is exceeded.

Example 3. *‘For any trips considered to have the ‘Bad’ condition, if that trip reaches a high level of tonnage and the work rolls recently had severe grindings in their last five trips, then the stopping of operation should be considered.’*

Here, we have to explicitly define ‘high tonnage’ and ‘severe grindings’ with the help of domain experts. In this case, high tonnage is a dynamic value that changes thresholds depending on the condition of the roll. Meanwhile, the condition of the work roll can be calculated by looking at the previous grindings and whether it has recently had any severe grindings. A roll grinding is considered severe if any of the last five grindings of the work roll had a stock removal value greater than 0.5 mm and therefore require a lower tonnage threshold. Additionally, the historical data of roll change reasons are also considered as part of the domain expert rule set. A severe grinding could be caused by a mill incident where the impact on the health of the roll is much greater. This knowledge is not captured in the RF but is included as expert rules. Listing 2 displays the rule in Example 3 in its SWRL format.

These rules are manually translated into the SWRL format. Afterwards, they were passed into the rule engine, which combined the RF prediction with the expert knowledge, producing new knowledge stored in the data property `hasStatus`. The status produced is displayed using SPARQL, a well-known semantic-based querying language that enables querying based on RDF triples and graph data [31]. Listing 3 is a SPARQL query, which when executed, prints the operational trips and their corresponding classification and status.

Listing 2. Example of a domain expert SWRL rule.

```
Roll(?r) ^ hasRollUnitTrip(?r, ?trip) ^ hasInferredCondition(?trip, ?condition) ^
swrlb:equal(?condition, "Bad") ^ hasRecentSevereGrinding(?r, ?severe) ^
swrlb:equal(?severe, true) ^ hasTripTonnage(?trip, ?tonnage) ^ swrlb:lessThan
(?tonnage, 2500) -> hasStatus(?trip, "Highest risk: predicted condition is
bad on a roll that was previously damaged. Recommended tonnage of 2500.")
```

Listing 3. SPARQL query to retrieve results.

```
PREFIX : <http://www.semanticweb.org/new/ontologies/2023/1/SCR0#>
PREFIX rf: <http://www.semanticweb.org/sadeer/ontologies/2023/0/RF#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX swrl: <http://www.w3.org/2003/11/swrl#>

SELECT ?trip ?classification ?status
WHERE {
  ?roll :hasRollUnitTrip ?trip .
  ?trip :hasRollRefrurbConditionInferred ?classification .
  ?trip :hasTripTonnage ?tonnage_of_trip .
  #if any of the last 5 trips contain a stock removal value >0.5
  ?roll :hasRecentHighStockRemoval ?high_stock_removal .
  ?trip :hasStatus ?status .
}
GROUP BY ?trip ?classification ?status
```

5. Results and Validation

The purpose of this section is to validate the framework by contextualizing it with the results obtained from the application, as illustrated in the use case presented in the preceding section. This validation process aims to assess the effectiveness and reliability of the proposed framework in practical scenarios, providing an evaluation of its performance and utility.

Figure 5 is a snippet that displays some results from the last 100 operational trips, which are used to test and validate the framework. These results display the knowledge-guided decisions through the Status column, as well as the classifications from semantic-based reasoning in the classification column, which are both displayed to the operator. As shown, each classification may be one of three categories, where the status provides a decision on what to do based on expert knowledge and the collected data at the specific timestamp of operation. The accuracy of the RF model is 78%, whereas the accuracy of the decision-guided assistance is calculated and validated with the help of domain experts in the following subsection using qualitative methods.

trip	classification	status
:roll_1647_trip_67061	"Bad"	"Stop rolling: predicted condition is bad on a roll that was previously damaged. Recommended tonnage of 2500 has been exceeded."
:roll_1647_trip_66936	"Best"	"Continue rolling: predicted condition is best but recent high stock removal may affect roll. Recommended tonnage of 4000."
:roll_1647_trip_66028	"Bad"	"Stop rolling: predicted condition is bad on a roll that was previously damaged. Recommended tonnage of 2500 has been exceeded."
:roll_1647_trip_65846	"Good"	"Stop rolling: predicted condition is good but recent high stock removal may affect roll. Recommended tonnage of 3500 has been exceeded."
:roll_1647_trip_65314	"Bad"	"Stop rolling: predicted condition is bad on a roll that was previously damaged. Recommended tonnage of 2500 has been exceeded."
:roll_1647_trip_64277	"Best"	"Continue rolling: predicted condition is best but recent high stock removal may affect roll. Recommended tonnage of 4000."
:roll_1647_trip_60465	"Best"	"Continue rolling: predicted condition is best but recent high stock removal may affect roll. Recommended tonnage of 4000."
:roll_1647_trip_59983	"Bad"	"Stop rolling: predicted condition is bad on a roll that was previously damaged. Recommended tonnage of 2500 has been exceeded."
:roll_1647_trip_59682	"Bad"	"Stop rolling: predicted condition is bad on a roll that was previously damaged. Recommended tonnage of 2500 has been exceeded."
:roll_1639_trip_60078	"Best"	"Continue rolling: predicted condition is best on a healthy roll. Recommended tonnage of 4500."
:roll_1639_trip_64659	"Bad"	"Stop rolling: predicted condition is bad on a previously healthy roll. Recommended tonnage of 3000 has been exceeded."
:roll_1639_trip_64367	"Good"	"Continue rolling: predicted condition is good on a healthy roll. Recommended tonnage of 4000."
:roll_1639_trip_60117	"Bad"	"Stop rolling: predicted condition is bad on a previously healthy roll. Recommended tonnage of 3000 has been exceeded."
:roll_1639_trip_59258	"Good"	"Continue rolling: predicted condition is good on a healthy roll. Recommended tonnage of 4000."
:roll_1639_trip_58274	"Bad"	"Stop rolling: predicted condition is bad on a previously healthy roll. Recommended tonnage of 3000 has been exceeded."
:roll_1639_trip_57965	"Bad"	"Continue rolling: predicted condition is bad on a previously healthy roll. Recommended tonnage of 3000."
:roll_1639_trip_57893	"Good"	"Continue rolling: predicted condition is good on a healthy roll. Recommended tonnage of 4000."
:roll_1639_trip_57416	"Bad"	"Stop rolling: predicted condition is bad on a previously healthy roll. Recommended tonnage of 3000 has been exceeded."
:roll_1638_trip_66551	"Bad"	"Stop rolling: predicted condition is bad on a previously healthy roll. Recommended tonnage of 3000 has been exceeded."
:roll_1638_trip_66441	"Best"	"Stop rolling: predicted condition is best on a healthy roll. Recommended tonnage of 4500 has been exceeded."
:roll_1638_trip_66127	"Good"	"Continue rolling: predicted condition is good on a healthy roll. Recommended tonnage of 4000."
:roll_1638_trip_66003	"Bad"	"Continue rolling: predicted condition is bad on a previously healthy roll. Recommended tonnage of 3000."
:roll_1638_trip_63403	"Bad"	"Continue rolling: predicted condition is bad on a previously healthy roll. Recommended tonnage of 3000."
:roll_1638_trip_62805	"Best"	"Continue rolling: predicted condition is best on a healthy roll. Recommended tonnage of 4500."
:roll_1638_trip_61717	"Best"	"Continue rolling: predicted condition is best on a healthy roll. Recommended tonnage of 4500."
:roll_1638_trip_60542	"Good"	"Continue rolling: predicted condition is good on a healthy roll. Recommended tonnage of 4000."

Figure 5. SPARQL query results.

Validation

The purpose and contribution of the framework is to provide improved assistance for the human in the loop. The classifications of the RF model produced good, bad, and best outputs, whereas the final status output provided new decision-making knowledge on whether to continue the rolling process, with an estimation of how much more rolling was recommended based on classification and expert knowledge.

Because of this, there was no ground truth or direct labels to compare for validation. However, there are many well-studied validation techniques for situations where no ground truth is available, often deployed during unsupervised learning models, such as interval validation, external validation, domain expertise, twin-sample validation, and cross-validation [37]. In our use case, we adopt external validation and domain expertise validation methodologies to validate our framework. These validation methods have been applied in various domains [38,39].

Similarly, we followed an iterative screening and refinement process with domain experts and stakeholders to share results and obtain valuable feedback, as highlighted in Figure 6. Within the iterative cycle, the results are displayed, discussed, and validated with the domain experts. The expert rules are refined with any newly obtained knowledge from these instances. Once the rule set is recompiled, it is passed through the semantic reasoner again, producing new results which are displayed to the experts once more, repeating the iterative cycle. By utilizing domain experts for validation, we can ensure the accuracy, relevance, and robustness of the framework, providing insights and tactic knowledge that are not apparent from data alone.

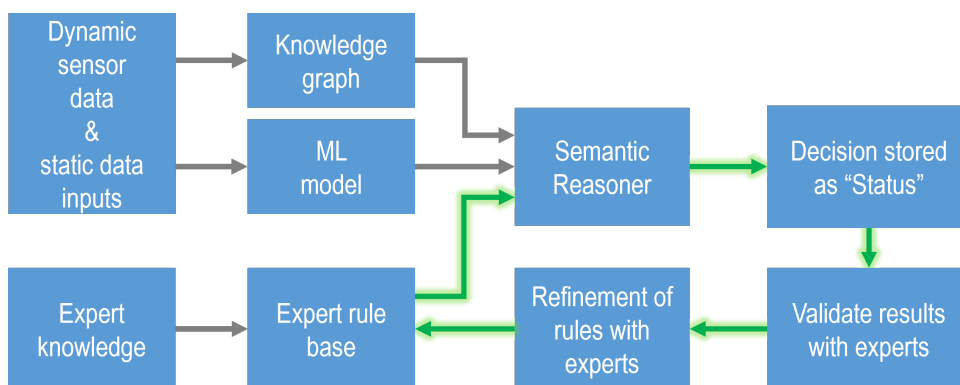


Figure 6. Methodology of iterative process with experts for validation.

With each iteration, we displayed the one hundred data entries. These entries contained the status, classification, expert rules, sensory data, and the actual stock reduction of the roll. Furthermore, we categorised the entries into two groups: (1) expected output consisting of 80% of the entries where the status of the entry matched the amount of stock reduction for that roll, e.g., “stop rolling” for entries where the stock reduction was greater than the expected average value, and (2) unexpected output where 20% of entries where the opposite interaction occurred. All entries were displayed to the experts; however, it would be revealed to be a time consuming task to iterate through all the entries, so we further refined and handpicked the most interesting results and compared the predictions with those of domain experts.

For the expected output category where the majority of data points were, ten entries were displayed to the experts. All entries were studied and approved by the experts, stating that they would have made the same decision based on the data. Two out of the ten entries are displayed in Table 2. This confirmed that the support provided by the framework was accurate for the expected category.

Similarly, ten entries from the unexpected output category were also displayed to the experts. In three of those cases, the status was ‘stop rolling’ despite a low diameter reduction and an accurate ‘good’ classification. In these cases, an expert rule was triggered

that prioritised stopping operation if there was a high tonnage produced on a pair of work rolls that were recently damaged. Having investigated these results with domain experts, we discovered that there was a ‘pinch’ on one of the rolls, which is required to be removed before further operation. This positive feedback confirmed that the framework was able to provide accurate results in these cases.

Table 2. Two reports from the ‘expected’ category.

Report ID	01
Roll and Trip ID	roll_1728_trip_67496
Category	Expected.
Trip details	(1) Trip has high tonnage threshold; (2) Work roll had a high stock reduction in the last five refurbishments.
Classification	Bad condition.
Status	Stop rolling: predicted condition is bad on a roll that was previously damaged. Recommended tonnage of 2500 is exceeded.
Expert comment	‘The status is accurate. The roll has done high tonnage in the current trip and should be taken out. Looking at the historical data of roll 1728, it had a significant cut in stock recently and should be treated carefully. If it was removed earlier, the stock reduction would be lowered.’
Report ID	02
Roll and Trip ID	roll_1609_trip_67249
Category	Expected.
Trip details	(1) Trip had medium tonnage; (2) Work roll is brand new and had no recently high stock reductions.
Classification	Best condition.
Status	Continue rolling: predicted condition is best on a healthy roll. Recommended tonnage of 4500.
Expert comment	‘The status is accurate to continue rolling. This roll is new and is expected to roll the maximum amount possible. The recommended tonnage could have been slightly higher than 4500 before being refurbished.’

Meanwhile, there were three instances where the status produced was ‘stop rolling’ by a similar expert rule. However, the experts stated that although the expert rule was correct, some knowledge was missing, stating that work rolls in Stand 3 are expected to handle more work load and pressure. The experts began to describe the refinement process in more detail, explaining how work rolls in different stands have different tonnage expectations and require different handling, which we adapt into our expert knowledge rule base for the next iteration. Finally, the remaining four instances provided an inaccurate status as the ML classification inaccurately predicted the roll condition. Two entries from the unexpected category are displayed in Table 3.

For the second iteration, we refined our expert knowledge rule set to include the newly obtained knowledge of stand information, which we displayed to the domain experts again. Once more, we produced two categories of results in the same manner as the first iteration. We first revisited the same entries from the first iteration. This time, the entries that were in Stand 3 produced a more accurate ‘continue rolling’ status, aligning with the expectations of the experts. Additionally, we continued to review ten new outputs for the second iteration for each category. This process established new knowledge regarding grinder stone diameter and roll hardness that can be used to improve the quality of the expert rules, followed by further iterations of the process if required.

From the qualitative validation, the accuracy of the expected category results for the first and second iteration were both 100%. Although each iteration had ten entries, it offered confidence to the experts using the system as 80% of the overall entries were in the expected category. Meanwhile, the accuracy of the unexpected column improved after the refinement of the expert rules in the second iteration, and it can be improved with further refinement.

Table 3. Two reports from the ‘unexpected’ category.

Report ID	03
Roll and Trip ID	roll_1534_trip_67845
Category	Unexpected.
Trip details	(1) Trip had high tonnage; (2) Work roll had a significant stock reduction in the last five refurbishments.
Classification	Good condition.
Status	Stop rolling: predicted condition is good but recent high stock removal may affect roll.
Expert comment	‘The status is accurate. After investigating roll 1534, there was a pinch on the work roll which needed to be removed before further operations.’
Report ID	04
Roll and Trip ID	roll_1647_trip_65846
Category	Unexpected.
Trip details	(1) Trip had high tonnage; (2) Work roll had a significant stock reduction in the last five refurbishments.
Classification	Good condition.
Status	Stop rolling: predicted condition is good but recent high stock removal may affect roll. Recommended tonnage of 3500 is exceeded.
Expert comment	‘The status is not completely accurate for this roll. This is because the roll is in stand three. Rolls in stand three are expected to do more total tonnage than other stands before being refurbished, and are expected to withstand stronger forces. The expected result in this situation is to continue rolling for a few hundred more tons.’

In addition, the experts we engaged with provided overwhelming positive feedback of the framework in general. One expert said that it is reassuring having a second opinion on difficult decision-making situations, where usually they may consult a fellow worker or manager for a second opinion. Meanwhile, another expert claimed that the framework has very good potential with the improvement of stronger expert rules and machine learning models. Finally, the iterative process itself was perceived as useful: it enabled interaction and continuous improvement of the decision making tool for the experts, while also providing us as non-experts with greater domain knowledge and understanding of the cold rolling processes.

Overall, the results demonstrated the capabilities of the framework, and its ability to assist operators with decision-making tasks. One fundamental contribution of the proposed framework is the ability to encode RF classification in a semantic way, so that the meta-data is also included in the model, which can be combined with external knowledge for further assistance in decision-making. Our results used real-world examples to demonstrate that such an approach is possible and advantageous. Ultimately, the knowledge is provided to assist and support the operator with their decision-making tasks.

6. Conclusions

This paper introduces a human-in-the-loop framework that combines a random forest model with semantic technologies where the resulting semantic ML classification is enhanced by domain expert knowledge. There are two key contributions and components of the framework: (1) the Random Forest Ontology that models the concepts of a random forest, which can be deployed and attached to an external ontology containing a knowledge graph to produce RF classification using rule-based reasoning, and (2) the integration of expert knowledge with the classification to provide semantically enriched, knowledge-guided decisions as recommendations for the human in the-loop system. A use case for predictive analytics in smart manufacturing is demonstrated that uses real-world data from industrial partners, which displayed the capabilities, applicability, advantages, and limitations of the proposed framework.

Within smart manufacturing, a fundamental goal is to improve machine interoperability and interpretability; this paper proposes a method of improving machine interpretability

via semantic technologies, providing one example of how ML models can be represented using semantic technologies.

7. Future Work

The major constraint of the proposed approach was the capabilities of the chosen rule engine. Drools, the default reasoner of SWRL-API, struggled to provide logical inferences on a very large dataset and hence limited the size of the knowledge graph. One future goal is to investigate the different available reasoners that have stronger inference capabilities to speed up and stabilise the inference process. On the other hand, the quality of the framework depends on the quality of the ML model and the quality of the training set, as well as the quality of the domain expert knowledge. Presently, we aim to hold more knowledge acquisition sessions to improve the quality of the expert rules. It is also necessary to take the time to evaluate and further validate the impact and effectiveness of the framework with the industrial partners after extended usage.

Author Contributions: Conceptualization, S.B.; Data curation, S.B. and K.L.; Formal analysis, S.B. and K.L.; Funding acquisition, C.G. and A.B.; Investigation, S.B.; Methodology, S.B.; Project administration, C.G. and A.B.; Resources, S.B., K.L. and C.G.; Software, S.B. and K.L.; Supervision, A.B.; Validation, S.B. and A.B.; Visualization, S.B. and A.B.; Writing—original draft, S.B. and K.L.; Writing—review and editing, C.G. and A.B. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by Swansea University and the Engineering and Physical Sciences Research Council grants EP/T517537/1, EP/V061798/1, EP/S001387/1 & EP/S018107/1.

Data Availability Statement: Restrictions apply to the availability of these data. Data was obtained from Tata Steel and are available from the authors with the permission of Tata Steel.

Acknowledgments: The authors would like to acknowledge Steve Thornton as the main contact from our industrial partners, who provided data and scheduled interviews with domain experts Mark and Chris. Finally, the authors would like to acknowledge Eugenio Borghini for his assistance with the RF model.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

API	Application Programming Interface
ML	Machine Learning
RDF	Resource Description Framework
RF	Random Forest
RFO	Random Forest Ontology
SCRO	Steel Cold Rolling Ontology
SPARQL	SPARQL Protocol and RDF Query Language
SQL	Structured Query Language
SWRL	Semantic Web Rule Language
W3C	World Wide Web Consortium

References

1. Patel, P.; Ali, M.I.; Sheth, A. From Raw Data to Smart Manufacturing: AI and Semantic Web of Things for Industry 4.0. *IEEE Intell. Syst.* **2018**, *33*, 79–86. [[CrossRef](#)]
2. Ameri, F.; Urbanovsky, C.; McArthur, C. A Systematic Approach to Developing Ontologies for Manufacturing Service Modeling. In Proceedings of the Workshop on Ontology and Semantic Web for Manufacturing, Graz, Austria, 24 July 2012.
3. Xiao, G.; Ding, L.; Cogrel, B.; Calvanese, D. Virtual Knowledge Graphs: An Overview of Systems and Use Cases. *Data Intell.* **2019**, *1*, 201–223. [[CrossRef](#)]
4. Wang, X.H.; Zhang, D.Q.; Gu, T.; Pung, H.K. Ontology based context modeling and reasoning using OWL. In Proceedings of the IEEE Annual Conference on Pervasive Computing and Communications Workshops, Proceedings of the Second, Orlando, FL, USA, 14–17 March 2004; pp. 18–22.
5. Dalzochio, J.; Kunst, R.; Pignaton, E.; Binotto, A.; Sanyal, S.; Favilla, J.; Barbosa, J. Machine learning and reasoning for predictive maintenance in Industry 4.0: Current status and challenges. *Comput. Ind.* **2020**, *123*, 103298. [[CrossRef](#)]

6. Xu, Y.; Sun, Y.; Liu, X.; Zheng, Y. A Digital-Twin-Assisted Fault Diagnosis Using Deep Transfer Learning. *IEEE Access* **2019**, *7*, 19990–19999. [[CrossRef](#)]
7. Roberts, W.L. *Cold Rolling of Steel*; Routledge: New York, NY, USA, 1978. [[CrossRef](#)]
8. Ray, A.; Mishra, K.; Das, G.; Chaudhary, P. Life of rolls in a cold rolling mill in a steel plant-operation versus manufacture. *Eng. Fail. Anal.* **2000**, *7*, 55–67. [[CrossRef](#)]
9. Rajbhandari, S.; Aryal, J.; Osborn, J.; Musk, R.; Lucieer, A. Benchmarking the applicability of ontology in geographic object-based image analysis. *ISPRS Int. J.-Geo-Inf.* **2017**, *6*, 386. [[CrossRef](#)]
10. Shoaip, N.; Rezk, A.; El-Sappagh, S.; Abuhmed, T.; Barakat, S.; Elmogy, M. Alzheimer’s Disease Diagnosis Based on a Semantic Rule-Based Modeling and Reasoning Approach. *Comput. Mater. Contin.* **2021**, *69*, 3531–3548. [[CrossRef](#)]
11. Jabardi, M.; Hadi, A. Twitter fake account detection and classification using ontological engineering and semantic web rule language. *Karbala Int. J. Mod. Sci.* **2020**, *6*, 8. [[CrossRef](#)]
12. Jabardi, M.; Hadi, A. Using Machine Learning to Inductively Learn Semantic Rules. In Proceedings of the International Conference of Modern Applications on Information and Communication Technology (ICMAICT), Babylon-Hilla City, Iraq, 22–23 October 2020; IOP Publishing: Bristol, UK, 2021; Volume 1804. [[CrossRef](#)]
13. Johnson, I.; Abécassis, J.; Charnomordic, B.; Destercke, S.; Thomopoulos, R. Making ontology-based knowledge and decision trees interact: An approach to enrich knowledge and increase expert confidence in data-driven models. In *Knowledge Science, Engineering and Management, Proceedings of the 4th International Conference, KSEM 2010, Belfast, UK, 1–3 September 2010*; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2010. [[CrossRef](#)]
14. Sarkar, A.; Naqvi, M.R.; Elmhadhbi, L.; Sormaz, D.; Archimede, B.; Karray, M.H. CHAIKMAT 4.0-Commonsense Knowledge and Hybrid Artificial Intelligence for Trusted Flexible Manufacturing. In *Flexible Automation and Intelligent Manufacturing: The Human-Data-Technology Nexus, Proceedings of the FAIM 2022, Detroit, MI, USA, 19–23 June 2022*; Springer: Berlin/Heidelberg, Germany, 2022; pp. 455–465.
15. Ammar, N.; Shaban-Nejad, A. Explainable artificial intelligence recommendation system by leveraging the semantics of adverse childhood experiences: Proof-of-concept prototype development. *JMIR Med. Inform.* **2020**, *8*, e18752. [[CrossRef](#)]
16. Bettini, C.; Civitarese, G.; Giancane, D.; Presotto, R. Procaviar: Hybrid data-driven and probabilistic knowledge-based activity recognition. *IEEE Access* **2020**, *8*, 146876–146886. [[CrossRef](#)]
17. Tofighi-Shirazi, R.; Asávoae, I.M.; Elbaz-Vincent, P. Fine-grained static detection of obfuscation transforms using ensemble-learning and semantic reasoning. In Proceedings of the 9th Workshop on Software Security, Protection, and Reverse Engineering, San Juan, PR, USA, 9–10 December 2019; pp. 1–12.
18. Pukkhem, N. A semantic-based approach for representing successful graduate predictive rules. In Proceedings of the 16th International Conference on Advanced Communication Technology, Pyeongchang, Republic of Korea, 16–19 February 2014; pp. 222–227. [[CrossRef](#)]
19. Cao, Q.; Zanni-Merk, C.; Samet, A.; Reich, C.; Beuvron, F.D.B.D.; Beckmann, A.; Giannetti, C. KSPMI: A Knowledge-based System for Predictive Maintenance in Industry 4.0. *Robot. -Comput.-Integr. Manuf.* **2022**, *74*, 102281. [[CrossRef](#)]
20. Maedche, A.; Staab, S. Ontology learning for the semantic web. *IEEE Intell. Syst.* **2001**, *16*, 72–79. [[CrossRef](#)]
21. Beden, S.; Cao, Q.; Beckmann, A. SCRO: A Domain Ontology for Describing Steel Cold Rolling Processes towards Industry 4.0. *Information* **2021**, *12*, 304. [[CrossRef](#)]
22. Chen, X.; Jia, S.; Xiang, Y. A review: Knowledge reasoning over knowledge graph. *Expert Syst. Appl.* **2020**, *141*, 112948. [[CrossRef](#)]
23. Xiao, G.; Lanti, D.; Kontchakov, R.; Komla-Ebri, S.; Güzel-Kalaycı, E.; Ding, L.; Corman, J.; Cogrel, B.; Calvanese, D.; Botoeva, E. The Virtual Knowledge Graph System Ontop. In *The Semantic Web—ISWC 2020, Proceedings of the 19th International Semantic Web Conference, Athens, Greece, 2–6 November 2020*; Pan, J.Z., Tamma, V., d’Amato, C., Janowicz, K., Fu, B., Polleres, A., Seneviratne, O., Kagal, L., Eds.; Springer: Cham, Switzerland, 2020; pp. 259–277.
24. Breiman, L. Random forests. *Mach. Learn.* **2001**, *45*, 5–32. [[CrossRef](#)]
25. Ali, J.; Khan, R.; Ahmad, N.; Maqsood, I. Random forests and decision trees. *Int. J. Comput. Sci. Issues (IJCSI)* **2012**, *9*, 272.
26. Altman, N.; Krzywinski, M. Ensemble methods: Bagging and random forests. *Nat. Methods* **2017**, *14*, 933–935. [[CrossRef](#)]
27. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.
28. Horrocks, I.; Patel-Schneider, P.F.; Boley, H.; Tabet, S.; Grosz, B.; Dean, M. SWRL: A semantic web rule language combining OWL and RuleML. *W3C Memb. Submiss.* **2004**, *21*, 1–31.
29. Beden, S.; Beckmann, A. Towards an Ontological Framework for Integrating Domain Expert Knowledge with RF Classification. In Proceedings of the IEEE 17th International Conference on Semantic Computing, Laguna Hills, CA, USA, 1–3 February 2023. [[CrossRef](#)]
30. Maarala, A.I.; Su, X.; Riekk, J. Semantic reasoning for context-aware Internet of Things applications. *IEEE Internet Things J.* **2016**, *4*, 461–473.
31. Pérez, J.; Arenas, M.; Gutierrez, C. Semantics and complexity of SPARQL. *ACM Trans. Database Syst. (TODS)* **2009**, *34*, 1–45. [[CrossRef](#)]
32. Schroeder, D.K.H. *A Basic Understanding of the Mechanics of Rolling Mill Rolls*; Eisenwerk Sulzau-Werfen—ESW: Werfen, Austria, 2003; pp.54–56.

33. Liang, S.; Fodor, P. OpenRuleBench: An Analysis of the Performance of Rule Engines. In Proceedings of the 18th International Conference on World Wide Web, Madrid Spain, 20–24 April 2009. [[CrossRef](#)]
34. Grosan, C.; Abraham, A.; Grosan, C.; Abraham, A. Rule-based expert systems. In *Intelligent Systems: A Modern Approach*; Springer: Berlin/Heidelberg, Germany, 2011; pp. 149–185.
35. Johnson, T.; Fletcher, S. How and why we need to capture tacit knowledge in manufacturing: Case studies of visual inspection. *Appl. Ergon.* **2019**, *74*, 1–9. [[CrossRef](#)]
36. Kidd, A. *Knowledge Acquisition for Expert Systems: A Practical Handbook*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2012.
37. Palacio-Niño, J.O.; Berzal, F. Evaluation metrics for unsupervised learning algorithms. *arXiv* **2019**, arXiv:1905.05667. <https://doi.org/10.48550/arXiv.1905.05667>.
38. Razavi, A.; Gill, H.; Ahlfeldt, H.; Shahsavar, N. Predicting Metastasis in Breast Cancer: Comparing a Decision Tree with Domain Experts. *J. Med. Syst.* **2007**, *31*, 263–273. [[CrossRef](#)]
39. Daneshjou, R.; Yuksekgonul, M.; Cai, Z.R.; Novoa, R.A.; Zou, J. SkinCon: A skin disease dataset densely annotated by domain experts for fine-grained debugging and analysis. In Proceedings of the Thirty-Sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track, New Orleans, LA, USA, 12–14 December 2022.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.