Enhancing car damage repair cost prediction: Integrating ontology reasoning with regression models

Hamid Ahaggach, Lylia Abrouk, Eric Lebon

Intelligent Systems with Applications 23 (2024) 200411

Contents lists available at ScienceDirect

Intelligent Systems with Applications

journal homepage: www.journals.elsevier.com/intelligent-systems-with-applications

Review

Enhancing car damage repair cost prediction: Integrating ontology reasoning with regression models

Hamid Ahaggach [a,c,*], Lylia Abrouk [a,b], Eric Lebon [c]

[a] LIB Laboratory, University of Burgundy, Dijon, France
[b] MISTEA, University of Montpellier, INRAE & Institut Agro, France
[c] Syartec, Aix-en-Provence, France

ARTICLE INFO

Keywords:
Cost prediction
Regression models
SWRL
Ontology reasoning
Named entity recognition
Relation extraction

ABSTRACT

The estimation of repair costs for car damage is a critical yet challenging task for insurance companies and repair shops. Accurate and the rapid predictions are essential for providing reliable cost estimates to customers. Traditional methods in this domain face multiple challenges, including manual processes and inaccuracies in repair cost estimation, as outlined in our article.

This paper introduces a novel approach that combines regression models with ontology reasoning to enhance the accuracy of car damage repair cost predictions. An Ontology for Car Damage (OCD)[1,2] has been developed, which is meticulously structured and populated using Named Entity Recognition (NER) and Relation Extraction (RE) techniques. This ontology provides a comprehensive framework for organizing and understanding the complex domain of car damage, capturing essential semantic relationships and variables that significantly influence repair costs. By integrating OCD with seven regression models, such as Random Forest and Decision Tree, we have proposed a hybrid methodology that leverages both structured data and semantic understanding. Our approach not only accounts for typical variables such as the type and severity of damage, and labor costs but also identifies novel features through the use of SWRL (Semantic Web Rule Language) rules, enhancing the model's predictive capabilities.

The performance of our models was evaluated using a substantial real-world dataset comprising over 300,000 records. This evaluation used metrics such as mean absolute error (MAE), root mean squared error (RMSE), and R-squared. The results indicate that our hybrid approach, which incorporates ontology reasoning, significantly outperforms traditional regression models.

The Random Forest model, especially when combined with the OCD ontology, showcased superior performance, exhibiting a minimal average deviation from the actual repair costs and achieving a low MAE.

This study's findings demonstrate the potential of combining ontology reasoning with machine learning techniques for precise cost prediction in the automotive repair industry. Our methodology offers a robust tool for insurance companies and repair shops to generate more accurate, reliable, and automated cost estimates, ultimately benefiting both businesses and customers.

Contents

* Corresponding author at: LIB Laboratory, University of Burgundy, Dijon, France.
E-mail addresses: Hamid.ahaggach@u-bourgogne.fr (H. Ahaggach), lylia.abrouk@u-bourgogne.fr (L. Abrouk), elebon@syartec.com (E. Lebon).

[1] industryportal.enit.fr/ontologies/OCD
[2] github.com/OntologyCarDamage/OCD

https://doi.org/10.1016/j.iswa.2024.200411
Received 23 November 2023; Received in revised form 10 April 2024; Accepted 17 June 2024
Available online 22 June 2024

1. Introduction and context

With the exponential growth of data in recent years, handling both structured and unstructured data has become essential, facilitating informed decision-making in various sectors (Lu, 2017). In the automotive industry, *Syartec*,[3] a leading software development company for car dealerships and insurance firms, plays a key role by offering comprehensive business management solutions that encompass everything from manufacturer sales to client deliveries.

A pressing challenge in this sector is the accurate assessment of car damage and the prediction of repair costs. Where insurance agents once traditionally inspected vehicles and documented damages in reports. These reports, often unstructured, necessitate manual data entry, a time-consuming and error-prone process. Furthermore, agents lack the means to estimate repair costs in real-time accurately. To address these issues, *Syartec* is pioneering the shift from manual to automated, reliable inspection protocols. However, evaluating car damage for repairs is fraught with challenges (Kyu & Woraratpanya, 2020; Martis, Sannidhan, Aravinda, & Balasubramani, 2023). The industry currently lacks standardized criteria for assessing damage based on type, severity, affected car parts, as well as the make and model of the vehicle. Moreover, the variability in part costs and labor charges, complicates this process.

Conventional methods (Department of Consumer Affairs Bureau of Automotive Repair 10949 North Mather Boulevard Rancho Cordova, 2022) for estimating repair costs often result in inconsistent and inaccurate figures due to the subjective nature of manual inspections and reliance on experience-based approximations. This inconsistency poses frustrations for insurance companies and customers alike, leading to potential financial discrepancies and delays in repairs. Consequently, there is a growing demand for more objective, data-driven approaches to automatically and efficiently estimate repair costs.

Numerous researchers (Kyu & Woraratpanya, 2020; Martis et al., 2023; Qaddour & Siddiqa, 2023; Sharma, Verma, & Gupta, 2019; Zhang et al., 2020; Zhu, Liu, Shen, & Zhao, 2021) have dedicated efforts to explore image-based methodologies for automating car repair cost estimation, utilizing advanced computer vision and machine learning algorithms. In parallel, industry experts (Inspektlabs, 2023; Ractable, 2023; Tchek, 2023) have also contributed significantly to this domain, applying similar technologies in practical, real-world applications. The process involves detecting the damaged car part, determining the type and severity of damage, and estimating the repair cost. However, image-based methods have several limitations. One of the major limitations is the complexity involved in accurately segmenting car parts in images. This is primarily due to the presence of numerous models and versions of vehicles, as well as constant updates to vehicle designs. Another limitation of image-based methods is that the model responsible for detecting damages can generate errors due to the reflection of light and the distance between the camera and the car. Additionally, detecting internal damages is challenging, and it is difficult to determine whether the damage requires replacement or repair.

On the other hand, approaches using tabular data to predict repair costs (Kim, Yum, Park, & Bae, 2021; Stojadinovic, Kovacevic, Marinkovic, & Stojadinovic, 2017), which leverage advanced statistical techniques and machine learning algorithms, have emerged as some of the most promising methods for repair cost estimation. These methods involve analyzing extensive datasets of tabular data to identify patterns and trends. Despite their potential, these approaches still face challenges. For instance, the accuracy of predictions is often limited by the quality of the data, which may be compromised by missing or incomplete information. Additionally, current methods do not fully capture the semantic information, and there is a notable gap in research specifically focused on predicting repair costs for car damages. These limitations highlight the need for a more sophisticated approach to car repair cost estimation that takes into account the semantic understanding of the different features involved.

The contributions of this article can be summarized as follows: (i) Presentation of OCD, an ontology for car damage, and ontology population using named entity recognition and relation extraction techniques. (ii) Definition and integration of semantic web rules for reasoning and enriching the features used by regression models to predict car repair costs. (iii) An innovative methodology that integrates ontology with regression models to enhance the accuracy of car damage repair cost predictions. (iv) Empirical validation of the proposed methodology through a comparative analysis of various regression models, both with and without ontology integration, using a real-world dataset.

[3] Syartec website: https://www.syartec.com.

The rest of the paper is organized as follows: Section 2 presents recent work on the use of regression models and ontologies and their various applications. This section also explores attempts to incorporate ontology into prediction models. Section 3 details the methodology for predicting car damage repair costs, outlining the various steps involved in the process. It also explains the integration of ontology to improve the accuracy of these predictions. In Section 4, a comparison of seven popular regression models is presented, both with and without the integration of ontology, using a real-world dataset. Illustrative examples for predicting car repair prices using an actual report case. The article concludes in Section 5 with a summary of the key findings, and Section 6 offers suggestions for future research directions.

2. Literature review

This section provides a comprehensive overview of existing studies using regression models and ontologies, highlighting their applications across various domains. The section also explores the advantages associated with using these techniques in forecasting damage repair costs. Additionally, it examines hybrid approaches that effectively combine machine learning with ontologies, detailing their effectiveness and potential in various applications.

In recent years, there have been significant advances in using machine learning algorithms, especially regression models, to estimate the cost of repair damage. These techniques can analyze large datasets and identify patterns and trends that help to estimate repair costs more accurately. Apart from using advanced ML techniques, the use of ontology can also be beneficial in predicting repair costs. Ontology is a powerful tool, enables the modeling of domain knowledge and facilitates the capture of semantic relationships between various features. The use of ontology can enhance the accuracy of cost predictions by introducing rules based on the Semantic Web Rule Language (*SWRL*). These rules enable the extraction of additional features and implicit knowledge, thereby improving the predictive capabilities of the model.

2.1. Regression-based approach

One approach to predicting repair costs is through the use of regression analysis (Bishop & Nasrabadi, 2006; Gareth, Daniela, Trevor, & Robert, 2013; Hastie, Tibshirani, Friedman, & Friedman, 2009), which is a statistical technique that models the relationship between a dependent variable, which in this case is the repair cost, and one or more independent variables, such as the characteristics of the damage and other relevant features. It involves estimating the parameters $\beta_0, \beta_1, \ldots, \beta_n$ of a linear or nonlinear equation of the form $y = f(x_1, x_2, \ldots, x_n) + \epsilon$, where ϵ is the error term (Lipatov, Belyanova, & Petunina, 2024). Regression models are used to predict future values of the dependent variable, identify important predictors, and understand the relationships between variables. However, in some cases, the relationship between the predictor variables and the response variable may be more complex and nonlinear, and in these cases, more advanced regression models may be used, such as polynomial regression (Hastie et al., 2009), support vector regression (Smola & Schölkopf, 2004), and neural network regression (Bishop et al., 1995). These advanced techniques can capture more complex patterns and relationships, and can result in more accurate predictions. Regression models have been applied to a wide range of applications in different fields, such as economics (Ye & Liu, 2022), finance (Cook, Kieschnick, & McCullough, 2008), construction (Jung et al., 2020) and healthcare (Kopitar, Kocbek, Cilar, Sheikh, & Stiglic, 2020; Stone, Zwiggelaar, Jones, & Mac Parthaláin, 2022).

In the automotive industry, regression analysis has been applied to a range of problems, such as estimating vehicle resale values (Gegic, Isakovic, Keco, Masetic, & Kevric, 2019; Lessmann & Voß, 2017), predicting a car's sale time (Ahaggach, Abrouk, Foufou, & Lebon, 2023), and maintenance prediction (Chen, Liu, Sun, Di Cairano-Gilfedder, & Titmus, 2019). Additionally, regression models have been used to estimate the costs in the automotive industry in many works. Huang, Huang, and Wu (2016) proposed the use of Partial Least Square Regression (*PLSR*) to estimate the cost of electric cars throughout their entire life cycle, considering features such as remaining mileage and battery capacity. Their study suggests that a reasonable choice of these features can improve the efficiency and reduce the overall cost of pure electric family cars. However, estimating the cost of manufacturing pure electric family cars presents challenges due to limited historical data availability and the collinearity of design parameters. *PLSR* is proposed as a solution to overcome these challenges and achieve more accurate cost estimations. Another study by Puripunyavanich, Myojo, and Kanazawa (2005) proposed a new method to estimate the lifetime maintenance and repair cost of durable goods, with a focus on automobiles in the US. The authors profiled the reliability characteristics of durable goods using statistical techniques and converted cross-sectional macro data on maintenance and repair expenditure per average household into longitudinal maintenance and repair cost per average good. The proposed statistical model can be applied to any consumer-oriented durable goods with significant mechanical components. In another work, Adekitan, Bukola, and Kennedy (2018) developed an artificial neural network (*ANN*) model to predict vehicle maintenance costs based on input data such as fuel volume, fuel cost, and car mileage. The study collected and analyzed data from two corporate organizations to identify common vehicle faults and their frequency of occurrence. The developed *ANN* model showed a significant correlation between the predictor inputs and the predicted maintenance cost. The model can be a useful tool for maintenance budget planning, as maintenance expenses make up a sizable portion of an organization's budget. The study suggests that the scope of research can be extended by collecting other parameters, both qualitative and quantitative, to improve the prediction model.

Moving forward, the subsequent section delves into an ontology-based approach, exploring relevant studies that have investigated the utilization of ontologies in prediction.

2.2. Ontology-based approach

This section reviews relevant studies that have explored the use of ontologies in the context of prediction. Ontology is a formal representation of concepts and relationships within a specific domain denoted by $\mathcal{O} = (\mathcal{C}, \mathcal{P}, \mathcal{J}, \mathcal{A})$, where $\mathcal{C}$ is a set of concepts, $\mathcal{P}$ is a set of properties, $\mathcal{J}$ of instances, and $\mathcal{A}$ is a set of axioms. It provides a structured and standardized way of representing knowledge, making it easier to share and reuse across different applications and systems. Specifically, in the context of prediction, ontology can be used to support the development of predictive models by providing a common vocabulary for describing the variables and relationships involved. The use of ontologies in prediction tasks has been explored in various fields such as construction, healthcare, seismic risk assessment, construction, failure classification, recommendation, and building cost estimation.

In the field of construction, ontologies have been used for automating the process of estimating construction costs. Niknam (2015) proposed a semantics-based approach to construction cost estimating using semantic web technology. The proposed approach uses ontologies to publish product information and develop ontology-based estimating applications. The semantic web services technology allows estimating applications to access the latest resource costs when needed, thereby eliminating the need for manual updating and improving estimator efficiency. Lee, Kim, and Yu (2014) proposed an approach for building cost estimation that uses building information modeling (BIM) data and ontological reasoning. The proposed approach emphasizes the use of BIM data to automate the search for work items suitable for building elements and materials. The proposed methodology can help to provide accurate and consistent results. Liu, Li, and Jiang (2016) proposed a method uses ontological modeling to represent cost estimation concepts

and their relationships, which are extracted from building specifications, construction documents, and BIM data. The proposed method can help to extract information more easily and quickly, thereby improving estimator efficiency. Hu and Liu (2020) proposed an e-maintenance platform design for public infrastructure maintenance based on IFC ontology and Semantic Web services. They estimated the cost based on a combination of BIM, international foundation for interoperable construction (IFC) data, and intelligent algorithms. The proposed methodology involves item extraction, quota standard selection, unit price analysis, work item pricing, and cost estimation

In healthcare, Thirugnanam, Thirugnanam, and Mangayarkarasi (2013) developed an ontology to offer accurate and relevant information about human diseases and their symptoms to users. They implemented *SWRL* rules for predicting diseases and performed various tests to ensure the proper functioning of the ontology. In another study (Chandra, Tiwari, Agarwal, & Singh, 2023), An ontology for vector-borne diseases was constructed, and *SWRL* was integrated for diagnostic and classification purposes.

In the field of seismic risk assessment, ontologies have been used to develop a holistic and probabilistic framework for assessing the risk of buildings during earthquakes. Xu et al. proposed an ontology-based holistic and probabilistic framework for seismic risk assessment of buildings (Xu, Zhang, Cui, & Zhao, 2022). The proposed method uses an Ontology-based Bayesian Belief Network and *SWRL* to determine the probability of a certain level of ground motion occurring within a specific time frame. The proposed method also helps to determine the total seismic risk probability of a structure, which can aid in making decisions about retrofitting or rebuilding the structure.

In recommendation systems, Fudholi, Maneerat, Varakulsiripunth, and Kato (2009) proposed a daily menu assistance system that suggests menus based on daily calorie needs. The system uses fuzzy ontology to provide menu recommendations based on factors such as price, rate, vote, and taste. The article explains the use of Protégé, *SWRL*, and *SQWRL* (Semantic Query-Enhanced Web Rule Language) in designing the recommendation feature and shows experimental results. The article also provides a brief review of fuzzy sets, fuzzy ontology, and daily menu assistance systems.

In fraud detection, Jabardi and Hadi (2021a) proposed an approach to detect fake Twitter accounts using ontology engineering and *SWRL* rules for scores calculation and fake Twitter detection was proposed. The authors inferred new features from given features and used them for classification. This article proposes an approach to detect fake Twitter accounts using ontology engineering and *SWRL* rules for scores calculation and fake Twitter detection. A new features are inferred from given features and can be used for classification. The paper also discusses the methodology, software, validation, formal analysis, investigation, resources, writing, and visualization.

The following section presents hybrid approaches that combine both ontology and machine learning techniques for various prediction tasks. Researchers have extensively explored this integration, aiming to leverage the strengths of both approaches to enhance the accuracy and effectiveness of predictions.

2.3. Hybrid approach

Many researchers have explored the integration of ontology and machine learning techniques for various prediction tasks. some works use the results of machine learning to construct rules that helps to improve the accuracy, some work only translates the rules found by ML algorithms into *SWRL* rules.

Cao, Samet, Zanni-Merk, de Beuvron, and Reich (2019) employed a combination of ontology and fuzzy clustering techniques to classify product failures. The authors utilized historical machine data to learn the criticality of failures through fuzzy clustering, and then used *SWRL* rules to predict the time and criticality of future failures based on the results of fuzzy clustering.

Tang, Liu, Yang, and Wei (2018) introduced a financial statement fraud detection system that utilized an ontology and a decision tree algorithm for fraud detection. The system combined decision tree rules to acquire rules *SWRL* and them to enable the inference engine to leverage existing knowledge and explore new knowledge.

In the same context, Jabardi and Hadi (2021b) use of machine learning and ontology to learn semantic rules in fraud classification. The proposed model uses an ontology to represent specific knowledge and decision trees as a data-driven rule learning method. They also discuss *SWRL* rules, which is used to mine hidden knowledge from huge ontologies.

In healthcare, Massari, Sabouri, Mhammedi, and Gherabi (2012) discuss the use of ML and ontology in predicting diabetes. In another work, El Massari et al. (2022) compared ontology on the prediction based on rules and machine learning algorithms to predict cardiovascular disease. In both works, the authors only translate decision tree rules to *SWRL* rules to predict the diabetes and cardiovascular disease. They conclude that ontologies yield better results when compared to the decision tree algorithms. This finding appears contradictory, since both methods rely on the same rules.

Tiwari, Chandra, and Agarwal (2022) proposed a methodology to predict the spread of COVID-19 using statistical and semantic web modeling techniques. They used the *ARIMA* model for time series forecasting and The *SWRL* rules are used for various purposes such as computing Body Mass Index (BMI), determining whether a patient is an adult or a minor, verifying the patient's gender, predicting COVID-19 cases using the *ARIMA* model, and calculating the probability of having COVID-19.

2.4. Discussion

Regression models are used to estimate repair costs for damage. These models require the analysis of extensive tabular data to discern patterns. However, despite their potential, there are challenges to address, particularly when dealing with low-quality data, missing or incomplete information, and complex patterns that complicate the task of capturing the relationship between features and the dependent variable.

The existing ontologies find application across various domains. However, it is worth noting that within the automotive domain, no specific ontology has been defined for car damage assessment. Furthermore, there is a dearth of research on prediction and car damage assessment within this context.

The integration of ontology and machine learning techniques has been widely investigated by researchers for various prediction tasks. Different approaches have been employed to leverage the benefits of machine learning and ontology in improving accuracy and rule generation. There are two main approaches observed in the literature when integrating ontology and machine learning techniques. Firstly, certain studies utilize the outcomes of machine learning to construct rules, aiming to enhance prediction accuracy. These rules are derived from the patterns and relationships identified by machine learning models and are designed to capture valuable insights, ultimately improving the overall performance of prediction tasks. Secondly, other research focuses on the translation of rules discovered by machine learning algorithms into *SWRL* rules. By converting the learned rules into *SWRL* format, these studies conclude that ontologies yield better results when compared to the ML algorithms such as decision tree algorithm. This finding appears contradictory, since both methods rely on the same rules.

Our approach combines ontology with regression models. By augmenting the model's input with additional characteristics and features derived from the ontology, the learning process is enriched, potentially uncovering new knowledge. Ontology reasoning is considered as a layer that contributes to improving prediction results. The next section provides a detailed description of this approach.

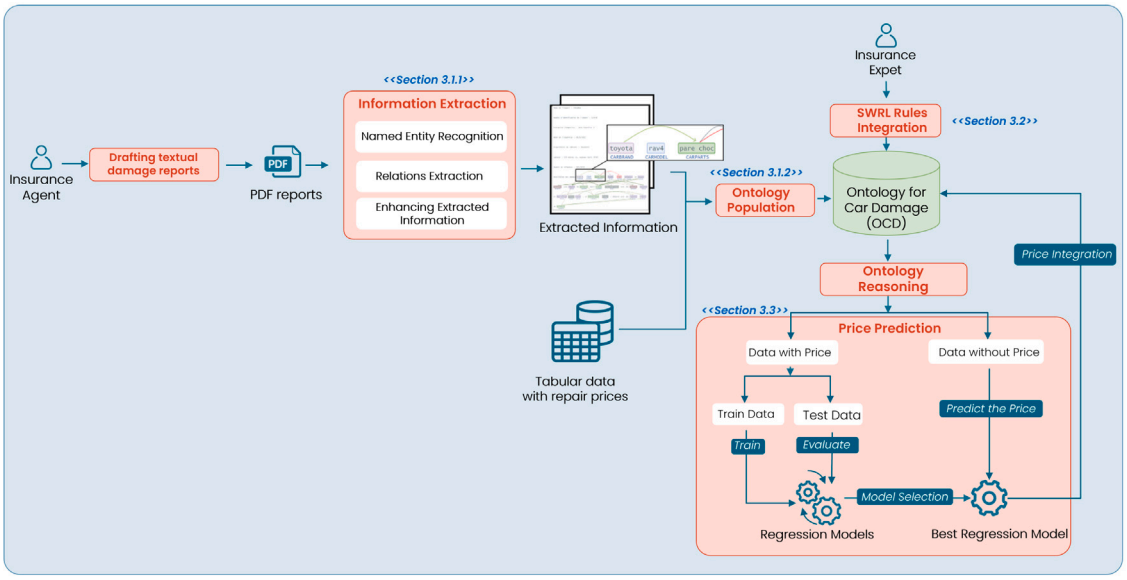

Fig. 1. Our methodology for car damage price prediction.

3. Proposed approach

This section details the methodology used to estimate car damage repair costs (see Fig. 1). The estimation process consists of three main phases: Information extraction for ontology population, SWRL rules integration, and price prediction.

In the information extraction for ontology population phase, pertinent information is extracted from unstructured damage reports using NER and RE techniques. This information is then populated into the proposed ontology. In the second phase, the integration of SWRL rules allows the ontology to perform reasoning that enhances and enriches this information. Lastly, the results from the ontology are used to predict the repair costs for each type of damage using regression model algorithms. Further details regarding our approach will be presented in the following section.

3.1. Information extraction for ontology population

Upon the arrival of each car, it undergoes a comprehensive quality control inspection. Any observed damage is carefully documented in an insurance report. However, these textual reports are unstructured and written in the French language, with each agent describing the damages in their own way.

The first step we completed in our previous work (Ahaggach, Abrouk, & Lebon, 2024) was to model the domain by constructing an ontology for car damage assessment. This ontology encompasses all the vocabulary, hierarchy of components, and different types of damage. We then extracted information using named entity recognition and relation extraction to retrieve relevant information from car damage reports and structured this information. Subsequently, we populated this information into the ontology.

3.1.1. Information extraction

Information extraction refers to the process of automatically extracting relevant information from text documents. This is typically accomplished through the application of natural language processing (NLP) and ML techniques. The extracted information can vary in complexity, ranging from simple facts such as names, dates, and locations to more intricate details like events, relationships, and sentiments. In our case, the goal is to extract entities and relations such as, car information (brand, model, color, etc.) entities, damage characteristics (severity, types, location, etc.), and the relations between these entities. For example, consider the following sentence: ***"Toyota Yaris with a minor dent on the back-left door and medium scratches on the bumper"***.

The extracted information would be as follows:

- ***Toyota*** as the brand of the car.
- ***Yaris*** as the model of the car.
- ***Door*** and ***Bumper*** as components of the car.
- ***Dent Scratches*** and ***Broken*** as types of damages.
- ***Severe*** and ***Medium*** as the severity levels of these damages.
- ***Back-left*** as the location of the damage.

All of these concepts are defined in the proposed ontology *OCD*. Additionally, we also extract relationships that exist between different entities to capture semantic information. For instance, in the previous sentence, the relation ***hasDamage*** exists between the car part *Door* and the damage type *Dent* to indicate that the door has a dent, as opposed to, for instance, the bumper, which is scratched.

Named entity recognition. NER stands as a pivotal task within the realm of NLP, centering on the identification and categorization of entities within textual data. The gamut of approaches for NER encompasses rule-based methods, dictionary-based methods, machine learning, and deep learning approaches. The overarching objective of NER revolves around labeling specific entities. In the NER task, our endeavor entails a comprehensive comparison of diverse machine learning algorithms, including Conditional Random Fields (Lafferty, McCallum, & Pereira, 2001), Bidirectional Long Short-Term Memory networks (Huang, Xu, & Yu, 2015), and FlauBERT[4] (Le et al., 2020), and SpaCy.[5] In our case, SpaCy NER model has proven its mettle by offering advanced capabilities for entity recognition.

Relation extraction. RE identify and extract the connections between entities mentioned in textual data. The objective is to establish connections between entities to capture the semantic information that enhances the comprehension of reported damages. During this task, we assess the performance of various machine learning models for the recognition and classification of relations. The considered models encompass support vector machines, K-nearest neighbors, decision trees,

[4] This model is available at: https://huggingface.co/flaubert/flaubert_base_cased.

[5] Spacy model: https://spacy.io/models/fr.

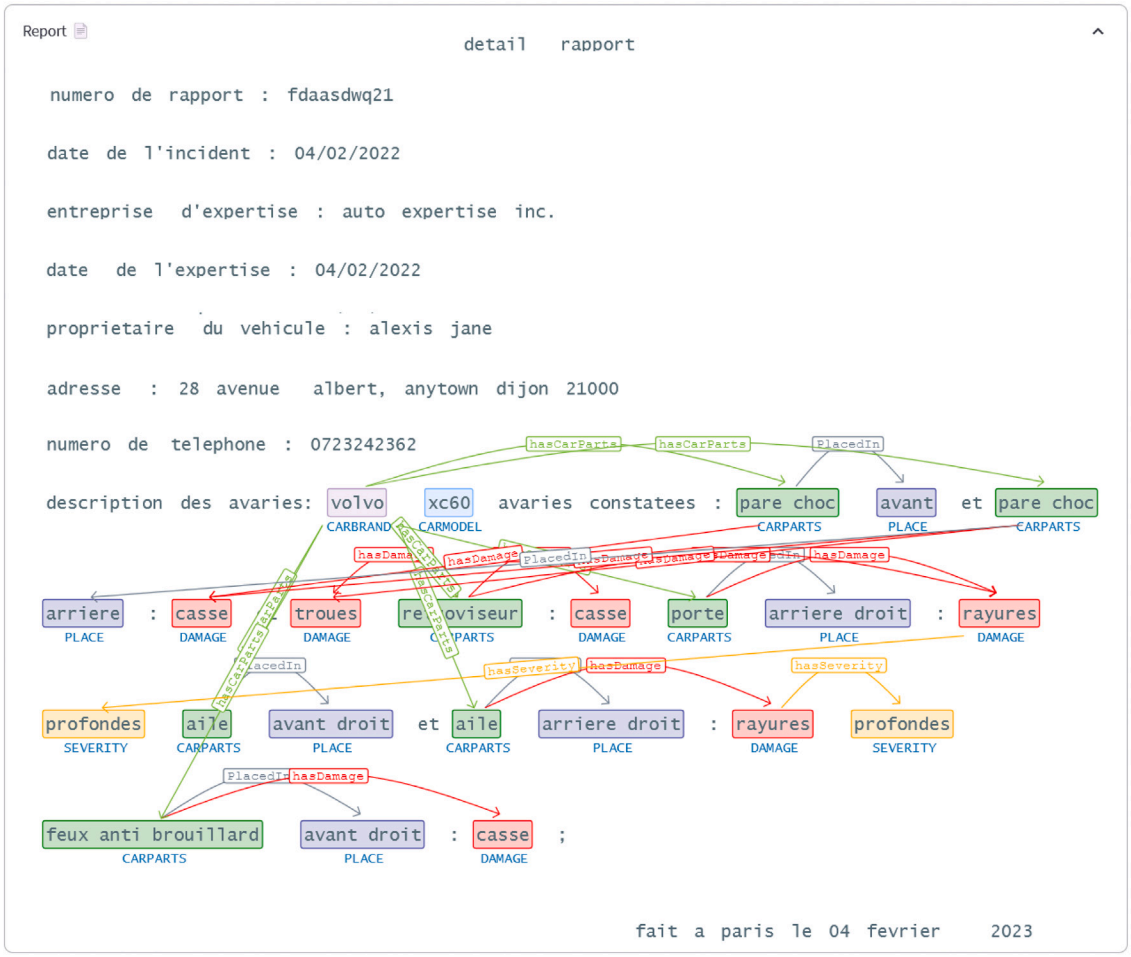

Fig. 2. Example of entities and relation extraction from a paragraph describing damage in a French language report.

and random forests. The primary aim is to identify the most effective model for relation extraction. Our evaluation results demonstrate that the random forests model outperforms the others, delivering superior performance within this specific context. All the comparison results of NER and RE are described in detail in our previous work (Ahaggach et al., 2024). The Fig. 2 illustrating an example of entities and relation extraction from a French report describing the car damages.

Enhancing extracted information. This step aims to improve the quality of the extracted entities and relations by reducing redundancy, resolving conflicts, and minimizing false positives and false negatives. It involves several subtasks, including:

(1) Deduplication: Identifying and merging duplicate or highly similar entities and relations to eliminate redundancy in the extracted information.
(2) Conflict Resolution: Resolving conflicting that may arise when multiple sources provide different information about the same entity or relationship. This process ensures consistency in the extracted data.
(3) False positive and false negative reduction: Addressing issues where incorrect entities and relations were extracted (false positives) and where relevant relations were missed (false negatives) during the extraction process. We use techniques such as threshold adjustment to filter information with low probabilities of belonging to entity or relation classes.
(4) Normalization: Standardizing extracted information to adhere to a consistent format in ontology.

3.1.2. Ontology population

This section presents the ontology for car damage (*OCD*), including its concepts, data properties, and object properties. Following that, we delve into the ontology population process and then discuss the evaluation process.

Ontology construction. The *OCD* ontology encompasses all aspects of car damage, including vocabulary, component hierarchy, and various types of damages. *OCD* has been developed based on the knowledge of car insurance experts and their descriptive reports. *OCD* serves the purpose of accommodating a wide spectrum of damage types and car models, effectively capturing pertinent data from damage description reports. Given the variability in the level of detail and information provided within these reports, the *OCD* ontology boasts a versatile and adaptable structure that facilitates seamless updates and modifications. This inherent flexibility renders it an invaluable tool for information extraction and analysis within the automotive sector, thus contributing significantly to the streamlining and optimization of the damage assessment and repair processes within the industry.

Ontology concepts. In our *OCD* ontology, we have established three fundamental concepts: *Damage*, *Car*, and *CarParts*. The concept of "*Damage*" serves as a broad category encompassing any form of harm, impairment, or loss inflicted upon a vehicle. Conversely, the "*Car*" concept represents any vehicle that undergoes transportation, whether via rail, truck, or ship. This encompasses new vehicles being transported from manufacturers to dealerships and used cars making their way to subsequent owners.

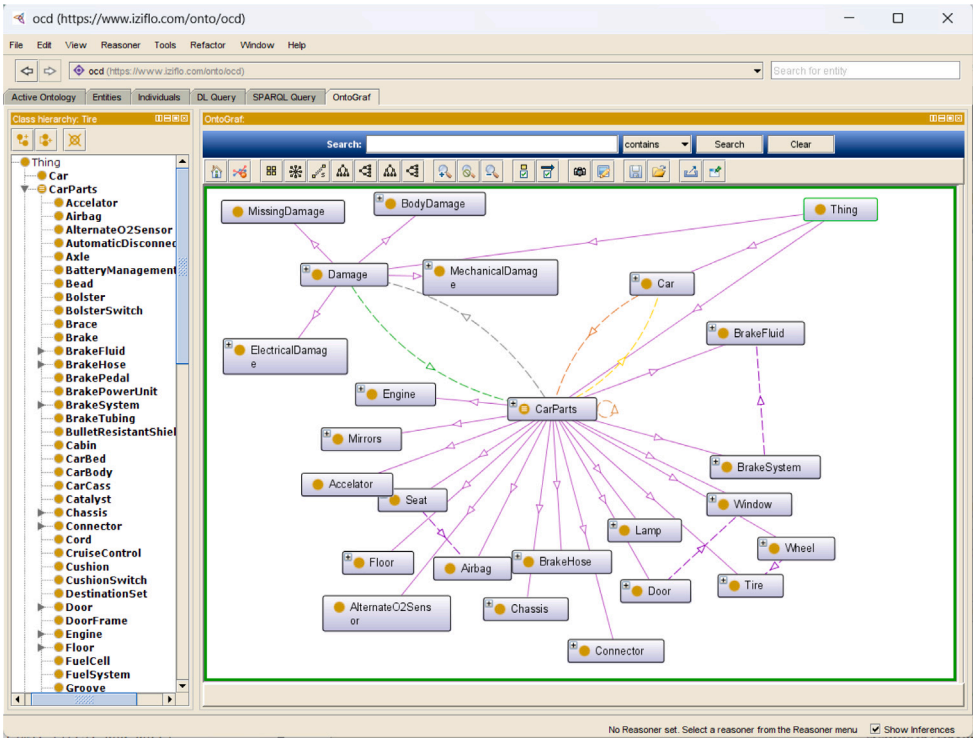

Fig. 3. A snapshot of the main classes present in the OCD ontology with protégé.

Moreover, the "*Damage*" concept can be more finely categorized into three primary subclasses:
"*BodyDamage*", "*MechanicalDamage*", and "*MissingDamage*".

- *Body Damage* : This refers to any harm or impairment that occurs to the external components of a car. This includes scratches, dents, cracks, and paint damage, among others. These types of damage are typically repaired by a body shop specialist with expertise in repairing and restoring car bodywork. Within the subclass of *Body damage*, we can identify different types of damage, such as superficial damage (minor scratches or dents that do not affect the car's structure), structural damage (major dents or impacts that affect the car's frame), and paint damage (discoloration, fading, or chipping of the car's paint).
- *Mechanical Damage* : This refers to any harm or impairment that occurs to the mechanical components of a car. This includes engine malfunctions, transmission issues, and brake system failures, among others. These types of damage are typically repaired by an auto mechanic who specializes in repairing and restoring the car's mechanical systems. Within the subclass of mechanical damage, we can identify different types of damage, such as electrical damage (faulty wiring or malfunctioning sensors), transmission damage (slipping gears or leaks), and engine damage (overheating or broken components).
- *Missing Damage* : This refers to any missing components or parts of the car. This can include items such as mirrors, headlights, or other exterior components, as well as interior components such as seats or radios. In some cases, missing damage may also affect the car's mechanical systems if significant components such as the battery or alternator are missing. Repairing missing damage typically involves replacing the missing components or parts with new ones, or in some cases, with used or refurbished ones.

Furthermore, some specific *CarParts* can be further broken down into multiple constituent parts. For instance, a car wheel is a type of *CarParts*, and it consists of several components, including wheel bearings, wheel rims, tires, and wheel fasteners. Fig. 3 provides a snapshot of the classes present in the ontology.

Data properties. The *OCD* ontology includes a number of data properties that can be used to describe various attributes of cars and car parts. Table 1 summarizes the data properties available in the *OCD* ontology.

Object properties. The *OCD* ontology includes several object properties that define the relationships between the classes. These object properties include:

- *hasCarPart*: This object property relates the car to its constituent parts. It has a domain of the class *Car* and a range of the class *CarParts*. For example, a car has an engine, wheels, seats, and so on. The inverse of this object property is *isPartOf*.
- *hasDamage*: This object property is used to define the relationship between a *CarParts* and *Damage*. The inverse of this object property is *inPart*.
- *hasComponent*: This object property establishes a relationship between a *CarParts* and its component *CarParts*. It denotes that a specific car part is composed of or includes other individual parts. This property is used to model hierarchical structures and compositions within the ontology, allowing for a detailed representation of the relationships between various car components.

Ontology population process. In this step, we focus on inserting the extracted information from unstructured French reports, along with structured data containing prices presented in Table 2 in our *OCD* ontology. To accomplish this task, we leverage the capabilities of the *Owlready*[6] package, which is a powerful tool for working with ontologies. *Owlready* provides a wide range of methods for efficiently handling ontologies, including the insertion of instances into the ontology. Our primary objective during this ontology population process is to map the extracted entities and relationships to the relevant properties and concepts within our ontology. This mapping ensures that the information derived from the reports becomes integrated seamlessly into the knowledge structure represented by the ontology.

[6] https://owlready2.readthedocs.io

Table 1
Data properties for the *OCD* ontology.

Data Property	Domain	Type	Description
CarBrand	Car	String	The company that produced the car
CarModel	Car	String	The model name of the car
CarYear	Car	Integer	The year the car was manufactured
CarColor	Car	String	The exterior color of the car
CarPrice	Car	Float	The price of the car in a given currency
CarRegistration	Car	String	The registration number of the car
FuelType	Car	String	The type of fuel, such as gasoline, diesel, or electric
CarMileage	Car	Integer	The total number of miles the car has traveled
PartName	CarParts	String	The name of the car part
CarPartMaterial	CarParts	String	The material(s) used to make the part
IsDamaged	CarParts	Boolean	This property specifies whether the car part is damaged or not
Place	CarParts	String	This property specifies the location of a damaged car part
PartPrice	CarParts	Float	The price of the part in a given currency
DamageType	Damage	String	The type of damage sustained by the car or car part
RepairCost	Damage	Float	The estimated cost of repairing the damage
Severity	Damage	String	The severity of the damage to the car part
RepairAction	Damage	string	Recommended repair action for the damage

Let us illustrate this process with an example: suppose we extract two entities from the reports, one representing a $CarParts$ and the other a $Place$, and these entities are linked by the relationship $PlacedIn$. In this scenario, the entity representing the $CarParts$ will be instantiated as a concept within the ontology, while the entity representing the $Place$ will be populated as a data property of the concept $CarParts$.

Ontology evaluation. Ontology evaluation is the process of assessing the quality of an ontology by measuring it against a set of established criteria, including accuracy, completeness, conciseness, adaptability, clarity, computational ability, and consistency. This helps to ensure that the ontology is reliable and can effectively support its intended applications (Raad & Cruz, 2015).

There are four common techniques used for evaluating ontologies (Asim, Wasim, Khan, Mahmood, & Abbasi, 2018; Hazman, El-Beltagy, & Rafea, 2011). The first technique, golden standard-based evaluation, compares the learned ontology to a standard one, representing the ideal knowledge representation for a specific domain. The second one is application-based evaluation, which focuses on assessing the ontology's performance in a particular task-specific application. The third one, data-driven or corpus-based evaluation, measures the ontology's coverage of a domain using domain-specific knowledge sources. Lastly, the expert-based evaluation usually involves evaluating the ontology through the experiences of users by defining indicators and assessing the ontology against each of them. In our case, we check the consistency of the ontology and ensuring that the reasoner does not produce any errors using the reasoners $Fact++$ (Tsarkov & Horrocks, 2006) and $HermiT$ (Shearer, Motik, & Horrocks, 2008).

We collaborated with car insurance experts to gather feedback and suggestions for improving our ontology. After conducting a series of interview sessions and analyzing sample data, we made several revisions to the ontology, including adding new classes and properties and refining definitions of existing ones in order to improve its effectiveness in representing and analyzing data. Furthermore, we assess the ontology's effectiveness by employing *SPARQL* (Pérez, Arenas, & Gutierrez, 2009). We also utilize it to extract data for training and evaluating our regression models designed to predict damage repair prices for instances without pricing information.

3.2. SWRL rules integration

This pivotal step effectively replaces the reliance on human experts, as it enables the addition of pertinent information, such as determining whether a damaged car part should be replaced or repaired based on a set of specific criteria established by the expert. By employing the *SWRL* rules, the ontology can identify components that can be reused, thereby reducing the necessity for new parts and consequently minimizing repair costs. *SWRL* rules also plays a crucial role in optimizing the repair price. For instance, this ontology incorporates a rule that stipulates if a damaged car door needs replacement and the car door handle is also damaged, there is no need to calculate the repair cost for the handle since it will be replaced alongside the door. In the following some examples of *SWRL* rules used in the ontology translated in English:

3.2.1. Rules decision

In this section, we have defined a set of rules to determine the appropriate repair action for damaged car parts based on various criteria. These rules provide a clear guideline for decision-making in the context of vehicle maintenance and repair.

- **Rule 1**: If the piece is damaged due to breakage, dent, or fracture, and it is a vehicle wheel, then the piece must be replaced.
  $\texttt{CarParts}(?part) \wedge \texttt{PartName}(?part, \texttt{"Wheel"}) \wedge \texttt{IsDamaged}(?part, True) \wedge \texttt{Damage}(?damage) \wedge \texttt{hasDamage}(?part, ?damage) \wedge (\texttt{DamageType}(?damage, \texttt{"Dent"}) \vee \texttt{DamageType}(?damage, \texttt{"}Fracture\texttt{"}) \vee \texttt{DamageType}(?damage, \texttt{"}Breakage\texttt{"})) \Rightarrow \texttt{RepairAction}(?damage, \texttt{"}Replace\texttt{"})$
- **Rule 2:** If the damage is related to car's bodywork (not wheels) and it is a dent with severity light, then perform the repair action on the damage.
  $\texttt{CarParts}(?part) \wedge \neg\texttt{PartName}(?part, \texttt{"Wheel"}) \wedge \texttt{IsDamaged}(?part, True) \wedge \texttt{Damage}(?damage) \wedge \texttt{hasDamage}(?part, ?damage) \wedge \texttt{DamageType}n(?damage, \texttt{"Dent"}) \wedge \texttt{Severity}(?damage, \texttt{"Light"}) \Rightarrow \texttt{RepairAction}(?damage, \texttt{"Repair"})$
- **Rule 3:** If a car part is damaged, including the wheel, and the damage type is $Broke$ regardless of the severity, the repair action is $Replace$.
  $\texttt{CarParts}(?part) \wedge \texttt{IsDamaged}(?\texttt{part}, True) \wedge \texttt{Damage}(?damage) \wedge \texttt{hasDamage}(?part, ?damage) \wedge \texttt{DamageType}(?damage, \texttt{"Broke"}) \Rightarrow \texttt{RepairAction}(?damage, \texttt{"Replace"})$
- **Rule 4:** If a car part is damaged and the damage type is $Fold$ and severity is $Light$, then the repair action is $Repair$.
  $\texttt{CarParts}(?\texttt{part}) \wedge \texttt{IsDamaged}(?\texttt{part}, True) \wedge \texttt{Damage}(?damage) \wedge \texttt{hasDamage}(?part, ?damage) \wedge \texttt{DamageType}(?\texttt{part}, \texttt{"Fold"}) \wedge \texttt{Severity}(?\texttt{damage}, \texttt{"Light"}) \Rightarrow \texttt{RepairAction}(?damage, \texttt{"Repair"})$
- **Rule 5:** If a car part is damaged, and the damage type is $Fold$ and severity is $Medium$ or $Strong$, then the repair action is $Replacement$.
  $\texttt{CarParts}(?\texttt{part}) \wedge \texttt{IsDamaged}(?\texttt{part}, \texttt{True}) \wedge \texttt{Damage}(?damage) \wedge \texttt{hasDamage}(?part, ?damage) \wedge \texttt{DamageType}(?\texttt{damage}, \texttt{"Fold"}) \wedge (\texttt{Severity}(?\texttt{damage}, \texttt{"Medium"}) \vee \texttt{Severity}(?\texttt{damage}, \texttt{"Strong"})) \Rightarrow \texttt{RepairAction}(?damage, \texttt{"Replacement"})$

Table 2
Tabular data containing repair cases with price information.

Report Ref	Car Brand	Car Model	Car Part	⋯	Place	Damage	Severity	Replace/Repair	Cost
vehicle_1171348_report	Honda	Civic	door	⋯	Front-right	Scratch	Minor	Repair	100 €
vehicle_1171443_report	Toyota	Camry	Rear Bumper	⋯	Rear	Dent	Major	Replace	500 €
vehicle_1171423_report	Ford	Mustang	Hood	⋯	Front	Crack	Major	Replace	1000 €
vehicle_1171348_report	Honda	Civic	Bumper	⋯	Front	Scratch	Minor	Repair	100 €
...	...	...	...	...	...	...	...	...	...
vehicle_1181441_report	BMW	3 Series	Windshield	⋯	Front	Chip	Minor	Repair	75 €

– **Rule 6:** If a car part is damaged and the damage type is *Scratch* for any severity type, then the repair action is *Repair*.
CarParts(?part) ∧ IsDamaged(?part, True) ∧ Damage(?*damage*) ∧ hasDamage(?*part*, ?*damage*) ∧ DamageType(?damage, "Scratch") ∧ Severity(?damage, ?severity) ⇒ RepairAction(?damage, "Repair")

3.2.2. Rules for knowledge discovery

In this section, we focus on knowledge discovery rules related to car damage and its potential impact on internal components. Here is an example:

– **Rule 7:** If a car has a damaged door, then there is a high probability that the internal components are also damaged if the severity of damage is *Strong*.
CarParts(?part) ∧ CarParts(?part2) ∧ IsDamaged(?part, True) ∧ Damage(?*damage*)∧hasDamage(?*part*, ?*damage*)∧DamageType(?part, "Scratch") ∧ PartName(?part, "Door") ∧ Severity(?damage, "Strong") ∧ hasComponent(?part, ?part2) ⇒ IsDamaged(?part2, True)

3.2.3. Rules for price reduction

In this section, we present an example of a rule that addresses price reduction considerations within the context of car part damage.

– **Rule 8:** If a car part is damaged, and the recommended repair action is replacement, and this car part has a damaged component, then the repair price of the component is zero, and the data property *IsDamaged* of the component is set to *False*.
−CarParts(?part) ∧ IsDamaged(?carPart, True) ∧ CarParts (?component) ∧ IsDamaged(?component, True) ∧ hasComponent (?part, ?component) ∧ Damage(?*damage*) ∧ hasDamage(?*component*, ?*damage*) ∧ RepairAction(?part, "Replace") ⇒ IsDamaged (?component, False)
−CarParts(?part) ∧ IsDamaged(?carPart, True) ∧ CarParts (?component) ∧ IsDamaged(?component, True) ∧ hasComponent (?part, ?component) ∧ Damage(?*damage*) ∧ hasDamage(?*component*, ?*damage*) ∧ RepairAction(?part, "Replace") ⇒ RepairCost (?damage, 0.0)

In Fig. 4, we illustrate the functionality of *SWRL* rules, with a particular focus on Rule 6, which add *RepairAction* information. This enriched data is subsequently employed in our regression models to forecast the repair costs of the damage. The predicted repair price is then seamlessly incorporated back into the ontology, ensuring that the ontology remains dynamically updated with the pricing information.

3.3. Price prediction

This section presents the prediction process after applying ontology reasoning, which allows us to perform preprocessing on unstructured report data and tabular. We execute *SPARQL* requests to extract data with prices (Table 2) for training and evaluating our regression models. Additionally, we extract data without prices to predict prices using the best regression model. In the next sections, we elaborate each step in this phase for estimating the price repair cost.

3.3.1. Dataset

We utilized a real dataset (Table 2) contains approximately 300,000 rows. This dataset comprises diverse information, including details about the car such as its brand and model, specifics about the damaged part requiring repair or replacement, as well as information concerning the severity and type of damage, along with the associated repair or replacement costs.

3.3.2. Data preprocessing

Before using the dataset, it is essential to preprocess it to ensure its quality and suitability for the task at hand. Data preprocessing involves a series of steps to clean, transform, and prepare the dataset. In our context, the following preprocessing steps are typically performed:

Handling missing data. This step is very important and sensitive because it directly affects the results of the model. In our case, we noticed missing data for several attributes. For instance, for attributes like *Severity*, we replaced the missing values with *Minor*, assuming that when severity information is missing, it can be considered as minor.

Data cleaning. Inspect the data for anomalies, outliers, and inconsistencies. For example in the *Cost* column, we have identified an anomaly where the cost is represented with the Euro symbol (€). To ensure consistency and facilitate numerical analysis, we should remove *e* from all entries in the *Cost* column. This step involves stripping the currency symbol and converting the column to a numerical format, making it suitable for further analysis.

Data encoding. To convert categorical variables (such as "Car Brand", and "Car Model") into numerical format, we use one-hot encoding and label encoding, depending on the nature of the data and the regression model to be applied.

Normalization and scaling. To ensure that both features are on a consistent scale, we apply Min–Max scaling. This scaling technique transform features into a standardized range, typically from 0 to 1, where 0 represents the minimum value in the original dataset, and 1 represents the maximum value.

Features selection. Before we delve into the details of the regression models, it is essential to carefully select the relevant features from our dataset for building accurate price prediction models. Feature selection is a crucial step that can significantly impact the performance of our models. The selected features should be informative and non-redundant to ensure the best possible prediction results.

The feature selection process involves identifying the most relevant attributes from the dataset that have a meaningful influence on the repair cost of car damage. We consider factors such as the car brand, car model, car part, location of the damage, severity of the damage, and whether repair or replacement is needed. To select the features, we employ Recursive Feature Elimination (RFE), Fig. 5 describe the process of *RFE*, this technique commonly used in machine learning, *RFE* operates by initially training a model with all available features, assessing their importance based on their contribution to model performance, and iteratively eliminating the least important ones until the model no longer improves. *RFE* is valuable for reducing dataset dimensionality, enhancing model efficiency, improving interpretability, and potentially boosting model generalization by concentrating on the most relevant information. However, its efficacy can vary depending on the specific dataset and modeling task, necessitating consideration of alternative feature selection methods as well.

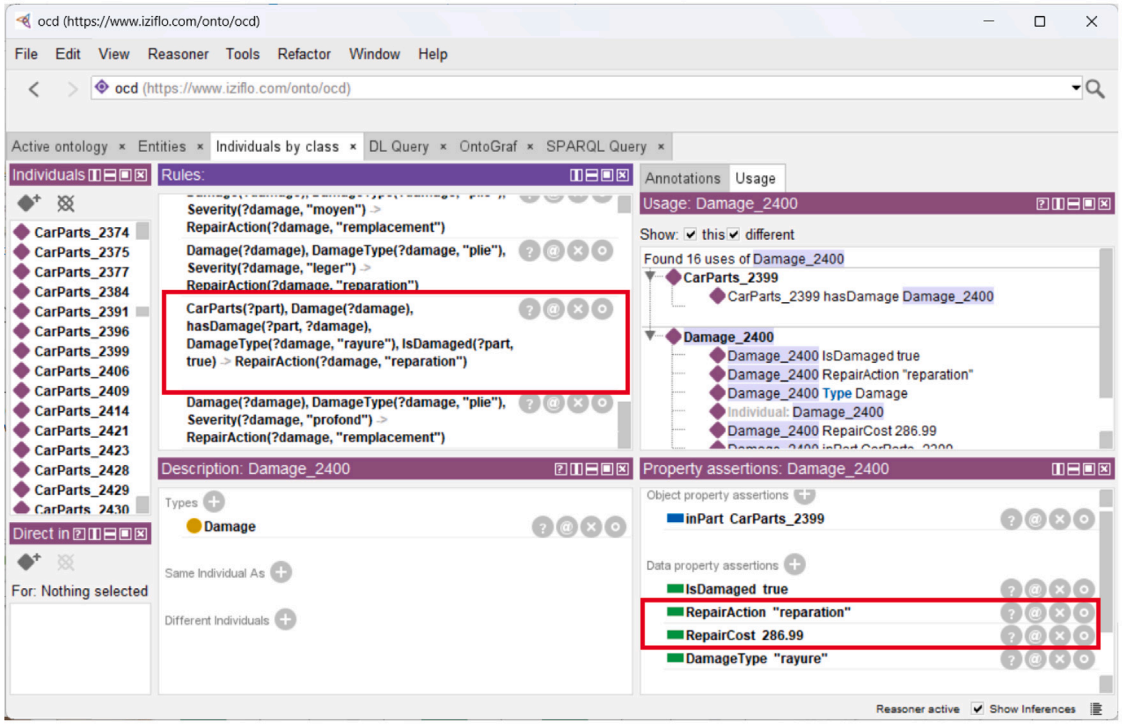

Fig. 4. Ontology reasoning with SWRL rules.

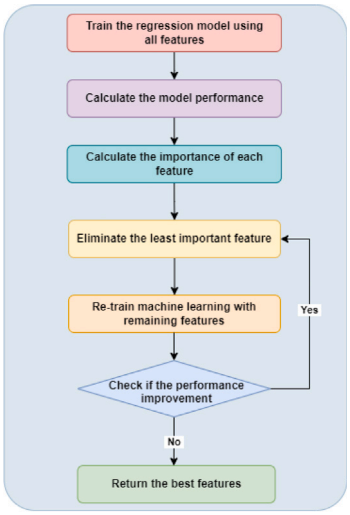

Fig. 5. Overview of recursive feature elimination for feature selection.

3.3.3. Regression models

In this section, we explore various regression models used to estimate the cost of car damage repair.

Let X be the set of input features in the dataset and features added by the ontology. Y is the target variable, the cost of repair or replacement. The dataset D, where $D = (x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$, $x_i \in X$ and $y_i \in Y$. Our goal is to learn a function $f : X \rightarrow Y$ that maps the input features to the cost of fixing the damage. To achieve this, various regression models such as linear regression, decision trees, and neural networks are used. Then, the most suitable model that yields the optimal prediction performance is identified.

Linear regression. Multiple linear regression (Galton, 1886) is used. It is an extension of linear regression that allows us to model the relationship between more than one independent variable and a dependent variable. The multiple linear regression equation in our case is as follows:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_n X_n + \epsilon$$

Where Y is the dependent variable, the cost of repair or replacement in euro. The coefficients β_0 to β_n represent the model parameters associated with independent variables X_1 to X_n, and ϵ represents the error term. This model allows us to estimate the cost of car damage repair based on the values of the independent variables. The optimization process seeks to determine the optimal values for the coefficients β_1 to β_n through training with data, with the objective of minimizing the sum of the squared differences between the actual and predicted prices.

Decision trees. A decision tree (Quinlan, 1986) is a tree-like model that represents decisions and their possible consequences. Each internal node of the tree represents a decision based on a feature, and each leaf node represents the prediction. Mathematically, a decision tree can also be represented as a function that maps input features (X) to the predicted repair cost (Y). Let us denote the decision tree as $f(X) = Y$. The decision tree algorithm learns to partition the feature space into regions, making predictions based on the average or majority label within each region. The algorithm determines the best feature and threshold to split the data at each node to maximize predictive accuracy.

Random forests. Random forests (Breiman, 2001) are ensemble learning models that combine multiple decision trees to make predictions. Each decision tree in the random forest is built independently using a different subset of the training data and a random subset of the features. The final prediction is obtained by aggregating the predictions of individual trees, either through averaging or voting.

Gradient boosting regressor. Gradient boosting regressor (Friedman, 2001) is a machine learning algorithm that combines multiple decision trees to create a strong predictive model. In each iteration, the

algorithm builds a new tree that corrects the mistakes made by the previous trees, with the objective of minimizing the overall prediction error.

XGB regressor. XGB Regressor (Chen & Guestrin, 2016) is an implementation of gradient boosting that utilizes the $XGBoost$ (eXtreme Gradient Boosting) framework. It is known for its high performance and efficiency in handling large-scale datasets. The XGB Regressor works similarly to the gradient boosting regressor by iteratively building decision trees and combining their predictions to make accurate estimates of the repair cost. It incorporates advanced techniques such as regularization and parallel processing to enhance model performance.

Support vector machines for regression (SVR). Support vector machines (Cortes & Vapnik, 1995) are supervised learning algorithms used for both classification and regression tasks. In regression, *SVR* works by finding a hyperplane in a higher-dimensional feature space that best fits the training data while controlling the margin of error. It can handle non-linear relationships between features and target values by using appropriate kernel functions.

Neural networks. This model is based on Multi-layer perceptron (Haykin, 1998). It uses an objective function, denoted as $f(x;\theta)$, where x signifies the input feature vector, and θ are the parameters of the neural network. This architecture enables a nonlinear transformation of the input features, effectively mapping them to the estimated cost. The neural network function can be represented as a composition of multiple layers, where each layer consists of a set of neurons that compute a weighted sum of their inputs and apply a nonlinear activation function to the result. The output of each layer is then used as input to the next layer until the final layer produces the predicted cost price.

Let x_i be the ith input feature of x, and let y be the true cost of repair or replacement. The neural network model is defined as follows:

$$y = f(x;\theta) + \epsilon,$$

where ϵ is the noise in the data and can be modeled as Gaussian noise with zero mean and variance σ^2.

To train the neural network model, a training dataset $D_{train} = (x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)$ is used, where N is the number of training examples. Our goal is to find the optimal parameters θ^* that minimize the difference between the predicted cost and the true cost price. This can be achieved by minimizing the mean squared error between the predicted cost and the true cost over the training dataset:

$$\theta^* = \arg\min_{\theta} \frac{1}{N} \sum_{i=1}^{N} (y_i - f(x_i;\theta))^2.$$

Backpropagation algorithm is used to compute the gradient of the *MSE* with respect to the parameters of the neural network, and then update the parameters using stochastic gradient descent algorithm (Ruder, 2016).

4. Experimentation

This section details the experimental analysis and evaluation used to assess the performance of various regression models for predicting the cost of car damage repairs. We explore the impact of incorporating an ontology on the prediction accuracy. The organization of this section is as follows: Initially, we describe the data partitioning and the computing environment used for model performance evaluation. We then discuss the evaluation metrics, selection of hyperparameters, and present the results. This includes a discussion of the findings and an illustrative example, concluding with a presentation of the limitations encountered.

4.1. Experimental setup

Our goal is to assess the performance of different regression models in predicting the cost of car damage repairs, both with and without the incorporation of an ontology. To conduct our experiments, the dataset (Table 2) of 300,000 rows, the dataset was partitioned into 70% for the training set, 10% for the validation set and 20% for the testing set, enabling us to train and evaluate the regression models. For model training, we employed a Dell $XPS159520$ laptop, equipped with an $12th$ Generation Intel(R) Core™ $i7-12700H$ processor with a 2.70 GHz clock speed, and running Windows 11 $Pro\ 64-bit$.

4.2. Evaluation metrics

Various metrics are used to evaluate the performance of regression models. These metrics provide a quantitative measure that allows us to compare different models and understand their strengths and weaknesses. Commonly used evaluation metrics in regression tasks are mean squared error, mean squared error, mean absolute error, and coefficient of determination. These metrics provide insight into the accuracy, precision, and variability of projected repair costs and ground truth values. below the equation of each evaluation metric:

$$\text{Mean Squared Error (MSE)} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

$$\text{Root Mean Squared Error (RMSE)} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2}$$

$$\text{Mean Absolute Error (MAE)} = \frac{1}{n}\sum_{i=1}^{n}|Y_i - \hat{Y}_i|$$

$$\text{Coefficient of Determination } (R^2) = 1 - \frac{\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2}{\sum_{i=1}^{n}(Y_i - \bar{Y})^2}$$

The *MSE* calculates the average of the squared differences between the actual repair costs (Y_i) and the predicted repair costs ($\hat{Y}_i$). It quantifies how far off, on average, the predictions are from the true values.

For *RMSE* is the square root of *MSE*. It represents the average magnitude of the errors in the same unit as the target variable.

The *MAE* calculates the average of the absolute differences between the actual and predicted repair costs. It measures the average magnitude of errors, regardless of their direction.

For R^2 coefficient quantifies the proportion of variance in actual repair costs explained by predicted repair costs, where Y_i represents the actual observed values of the dependent variable. $\hat{Y}_i$ represents the predicted values of the dependent variable obtained from the regression model. $\bar{Y}$ represents the mean of the observed values of the dependent variable. The higher values indicate better model performance in capturing relationships between features and repair costs.

There are alternatives to these metrics, such as Mean Percentage Error (MPE), Mean Absolute Percentage Error (MAPE), Median Absolute Deviation (MAD), and Adjusted R-squared. However, these alternatives were not selected as the primary metrics because the initial four metrics sufficiently cover the necessary aspects of model evaluation for our purposes. In addition, the alternatives present specific limitations: MPE can result in misleading conclusions in the presence of zero actual values, and is less effective where the magnitude of absolute errors is critical. MAPE is influenced by zero actual values and may disproportionately penalize underpredictions, possibly conflicting with our study's objectives. MAD provides a robust variability measure, but not fully capture error magnitudes crucial for cost estimation. Adjusted R-squared offers insights into the model's explanatory power adjusted for the number of predictors. However, the initially selected metrics adequately assess model performance for our purposes.

In addition, we extend the evaluation by comparing the average prediction time of the two methods, with and without ontology integration. This comparison aims to quantify the impact of ontology on

Table 3
Optimal hyperparameters for regression models.

Model	Parameter	Definition	Optimal Value
Linear regression	fit_intercept	Specifies if the constant intercept added to the decision function.	True
	normalize	Specifies if the predictors X normalized before regression.	True
Decision tree	criterion	The function used to measure the quality of a split.	MSE
	max_depth	The maximum depth of the tree.	10
	min_samples_split	The minimum number of samples required to split an internal node.	4
	min_samples_leaf	The minimum number of samples required to be at a leaf node.	2
Random forest	bootstrap	Whether bootstrap samples are used when building trees. If False, the whole dataset is used to build each tree.	False
	max_depth	The maximum depth of the tree. If None, then nodes are expanded until all leaves are smaller than min_samples_split samples.	None
	max_features	The number of features to consider when looking for the best split.	auto
	min_samples_leaf	The minimum number of samples required to be at a leaf node.	1
	min_samples_split	The minimum number of samples required to split an internal node.	2
	n_estimators	The number of trees in the forest.	100
SVR	C	Regularization parameter. The strength of the regularization is inversely proportional to C.	10
	epsilon	Epsilon defines a margin of tolerance where no penalty is given to errors.	1
	kernel	Specifies the kernel type to be used in the algorithm.	rbf
Gradient boosting	learning_rate	Learning rate shrinks the contribution of each tree by learning_rate.	0.1
	max_depth	Maximum depth of the individual regression estimators.	3
	min_samples_leaf	The minimum number of samples required to be at a leaf node.	1
	min_samples_split	The minimum number of samples required to split an internal node.	2
	n_estimators	The number of boosting stages to be run.	100
MLP	hidden_layer_sizes	The ith element represents the number of neurons in the ith hidden layer.	100
	activation	Activation function for the hidden layer.	relu
	solver	The solver for weight optimization.	adam
	alpha	L2 penalty (regularization term) parameter.	0.001
	learning_rate	Learning rate schedule for weight updates.	0.01
	batch_size	Refers to the number of training examples utilized in one iteration of model training.	32
XGB	n_estimators	Number of gradient boosted trees. Equivalent to number of boosting rounds.	100
	max_depth	Maximum tree depth for base learners.	6
	learning_rate	Boosting learning rate.	0.1
	subsample	Subsample ratio of the training instances.	0.8
	colsample_bytree	Subsample ratio of columns when constructing each tree.	0.8

computational efficiency. By measuring the time taken for each model to predict repair costs on a standardized test set. This aspect of the evaluation is crucial for practical applications, where both accuracy and speed are important factors in the deployment of models.

4.3. Hyperparameter selection

To ensure optimal performance of regression models, choosing appropriate hyperparameters is necessary. Hyperparameters are configuration settings set before a model is trained, rather than learned from data. Experiments explore different combinations of hyperparameters and evaluate model performance using the grid search technique. All hyperparameters used are detailed in the Table 3.

4.4. Results

The Table 4 presents a comprehensive comparison of various regression models, evaluated both with and without the use of ontology. The models are assessed based on four metrics: *MSE*, *MAE*, *RMSE*, and R^2. In the scenario without ontology, the *MLPRegressor* model exhibits the best performance with an R^2 score of 91%, indicating an acceptable fit to the data. Conversely, the *SVR* model demonstrates the lowest performance with an R^2 score of 61%. This suggests that, without ontology, *MLPRegressor* is a more reliable choice for this dataset because it better captures the underlying patterns in the data. This is due to the presence of numerous feature transformations within its layers, allowing it to make more accurate predictions.

The integration of ontology into the regression models resulted in significant enhancements in prediction accuracy for all models. The *RandomForestRegressor* and *DecisionTreeRegressor* models emerged as the top performers, providing the most precise cost estimates for car damage repair following ontology integration.

The performance of the *RandomForestRegressor* model significantly improved when ontology was integrated, achieving a low mean squared error of 114 and a low mean absolute error of 2 this indicates that, on average, the predicted repair costs deviate from the actual costs by only 2 euros. It outperformed all other models, including those without ontology. The high R^2 value of 97% indicates that the predicted cost estimates closely align with the actual values. The model's ability to capture complex relationships and patterns within the data makes it a strong candidate for accurate cost estimation. Similarly, the *DecisionTreeRegressor* model also exhibited outstanding performance with an MSE of 138 and an R^2 value of 96%. As a tree-based model, it is capable of learning intricate non-linear relationships between the input features and the target variable. The model's flexibility and ability to capture complex interactions within the data contribute to its predictive power.

Although the *MLPRegressor* and *GradientBoostingRegressor* models exhibited relatively higher *MSE* values compared to RandomForestRegressor and *DecisionTreeRegressor* models, they still provided reliable cost estimates. The *MLPRegressor* model achieved an *MSE* of 206 and an R^2 value of 94%, indicating strong predictive capability. The *GradientBoostingRegressor* model achieved an *MSE* of 368 and an R^2 value of 89%. On the other hand, the *SVR* and *LinearRegression* models showed relatively weaker performance in both scenarios, with higher *MSE* values and lower R^2 values. The *SVR* model achieved an *MSE* of 1214 and an R^2 value of 64%, while the *LinearRegression* model had an *MSE* of 918 and an R^2 value of 73%. It is noteworthy that the inclusion

Table 4
Regression model comparison with and without ontology incorporation.

Model	Without Ontology				With Ontology			
	MSE	MAE	RMSE	R2	MSE	MAE	RMSE	R2
LinearRegression	973.208	22.720	31.196	0.712	918.140	22.035	30.300	0.727
DecisionTreeRegressor	769.082	8.600	27.732	0.772	138.071	2.133	11.750	0.959
RandomForestRegressor	692.443	8.517	26.314	0.795	**114.126**	**2.043**	**10.683**	**0.966**
GradientBoostingRegressor	723.964	18.313	26.906	0.786	367.824	13.482	19.178	0.890
XGBRegressor	936.508	21.616	30.602	0.723	662.352	18.088	25.736	0.803
SVR	1330.991	21.927	36.482	0.607	1214.532	20.198	34.850	0.640
MLPRegressor	285.921	10.334	16.909	0.915	206.529	7.482	14.371	0.938

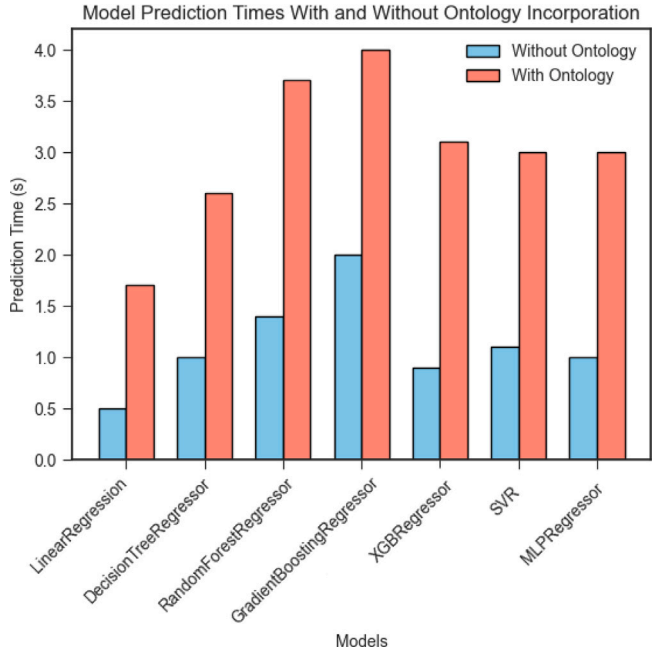

Fig. 6. Comparison of the average prediction time of regression models with and without the integration of ontology.

of ontology significantly improved the performance of all regression models tested.

As illustrated in Fig. 6, the bar chart presents a comparative analysis of average prediction times for regression models with and without the incorporation of ontology. It indicates that for every model, the inclusion of ontology increases the prediction time. This increase is likely due to the additional computational overhead introduced by ontology processing, which involves population, reasoning and handling structured knowledge that is not present when the models operate without ontology.

Models like *RandomForestRegressor* and *GradientBoostingRegressor* show a substantial increase in prediction time with and without ontology. This is because these models are ensemble methods involving multiple decision trees.

The *XGBRegressor*, *SVR* and *MLPRegressor* also show increased prediction times with ontology, but the increase is less pronounced compared to *RandomForestRegressor* and *GradientBoostingRegressor*. This is explained by the fact that these models, although complex, are better optimized for handling the structured input data, or the ontology integration is done in a way that does not add as much computational burden relative to the other models. *LinearRegression* shows the least increase in prediction time with the incorporation of ontology. Since this model is generally simpler in nature, the ontology's structured data do not add as much complexity to the prediction process as it does with more complex models.

4.4.1. Results discussion

The remarkable improvement in prediction accuracy (Fig. 7) for all models with the integration of ontology attributed to several factors. Ontologies, by design, enhance the semantic richness of the data, providing more features and knowledge in the domain of car damage repair cost estimation. This structured representation enables models to better understand the relationships and attributes of entities within the data, leading to more accurate predictions.

The superior performance of the *RandomForestRegressor* compared to the *DecisionTreeRegressor* model is attributed to the fact that *RandomForest* contains several decision trees that collaborate among themselves, making decisions based on the majority vote. *RandomForest* employs a bagging technique that aggregates multiple decision trees to reduce variance without increasing bias. This characteristic makes it less prone to overfitting the data compared to simpler models, contributing to its higher accuracy and lower error rates. Furthermore, the model's ability to handle non-linear relationships and interactions between variables is significantly enhanced by the integration of an ontology. This allows it to leverage structured data to identify more complex patterns, further improving its performance.

Similarly, the *DecisionTreeRegressor* benefits from the ontology's structured data, which aids in the decision-making process at each node of the tree. This allows the model to make more informed splits that closely represent the underlying data structure. The high R^2 value indicates that the model is capable of capturing a significant portion of the variance in the data, which is crucial for accurate predictions.

The relatively poorer performance of the *MLPRegressor*, *GradientBoostingRegressor*, *SVR*, and *LinearRegression* models are attributed to different factors. The *MLPRegressor*, a type of neural network, and the *GradientBoostingRegressor*, which is an ensemble of decision trees that uses boosting, are complex models that can capture intricate patterns in the data. However, their performance may be hindered by the need for extensive parameter tuning and the risk of overfitting, especially in cases where the data is not sufficiently large or diverse. Despite these challenges, their predictive capabilities remain strong, as indicated by their R^2 values, showing that they are still capable of capturing a significant amount of variance in the target variable. The *SVR* and *LinearRegression* models, being simpler and more linear in nature, struggle with the complex and non-linear relationships that are present in the car damage cost estimation data. Their lower performance metrics suggest that these models are less capable of handling the complexity and diversity of the data, even with the semantic enhancement provided by ontology. The high *MSE* values and lower R^2 values reflect their limited ability to accurately predict car damage repair costs.

The integrating ontology into regression models significantly enhances their prediction accuracy by providing a richer semantic understanding of the data. However, this integration also increases the prediction time. Models like *RandomForestRegressor* and *DecisionTreeRegressor* excel in this enriched data environment due to their ability to handle complexity and non-linearity, making them particularly suited for tasks like car damage cost estimation. The varying performance among the models underscores the importance of selecting the appropriate model based on the specific characteristics of the data and the task at hand.

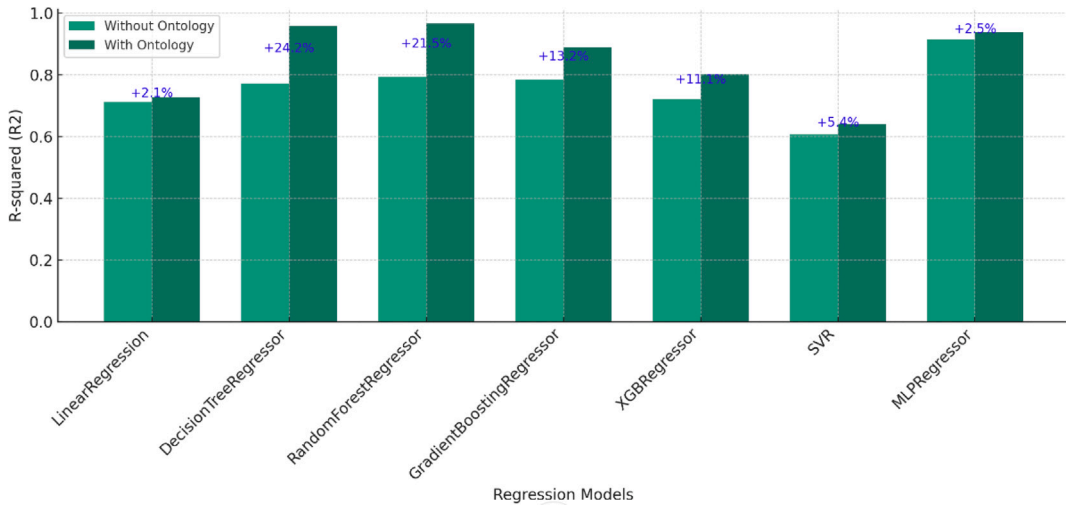

Fig. 7. Comparison of regression models by R^2 with percentage improvement.

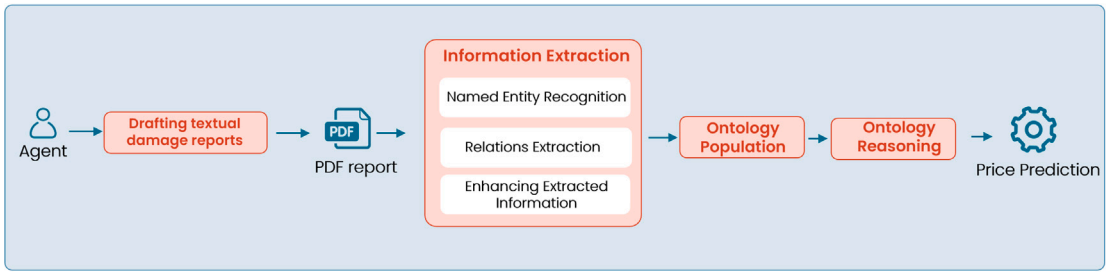

Fig. 8. Illustrative process of our methodology's use case for predicting car damage cost based on PDF reports.

4.5. Illustrative example

This section illustrates a practical application of our methodology through examples and discuss scenarios where our method may not perform optimally. As depicted in Fig. 8, the process for predicting the repair or replacement cost for each type of damage begins with the analysis of the textual report provided by the agent. Subsequently, the report is processed to extract entities and their relationships, thereby capturing the semantic information. Following this, a post-processing step is undertaken to refine the extracted information. This refined information is used to populate the ontology, which includes SWRL rules. Reasoning is initiated to execute the rules outlined in Section 3.2, thereby generating additional features. These features are then fed into the best regression model to estimate the repair or replacement costs for each type of damage. Fig. 9, illustrate comparison of actual and predicted car damage costs for 20 cases using random forest in conjunction with ontology cooperation. We detailed three specific cases in 5 where the actual price differs from the predicted price. This table presents the input text of the damage description, the information extracted, the outcome of the ontology reasoning, the predicted, and the actual price, along with the variance between the predicted and actual costs.

Case #1 (Toyota Yaris): The methodology successfully identifies the damage and calculates a total predicted price of 1124€, which is very close to the actual repair cost of 1122€. This example illustrates the accuracy of the model in scenarios with precise damage descriptions.

Case #12 (Ford Focus): The model predicts a total repair cost of 906€, compared to the actual cost of 937€, resulting in a variance of 31€. The slight discrepancy is due to the entity "deep" being missed by the NER model. This oversight in assessing severity led the ontology to decide on repair only, instead of replacement.

Case #14 (Volvo XC60): The model predicted a cost of 1104€, which is higher than the actual cost of 1061€. This difference is attributed to the ontology decision to replace parts which is correct; however, the issue arises from the regression model, which assumes the price of new parts. In reality, the replacement was made using second-hand parts, leading to a lower actual cost.

4.6. Limitations

This study's methodology, while innovative, encounters specific limitations. A notable concern is the prediction time when integrating ontology with regression models. This incorporation potentially increases computational complexity, leading to longer processing times due to the additional steps in ontology reasoning. However, for insurance companies, the trade-off for greater accuracy in cost prediction is deemed more critical, underscoring the industry's priority on precision over speed.

Another critical aspect is the accuracy of predictions dependent on named entity recognition and relation extraction techniques. Inaccuracies in these initial steps could potentially impact the model results. To mitigate the impact of inaccuracies, an interface has been implemented for experts to modify relationships and entities as needed. This solution not only addresses concerns of inaccuracies but also contributes to the model's accuracy.

5. Conclusion

In addressing the challenges faced by insurance companies and repair shops in estimating car damage repair costs accurately and

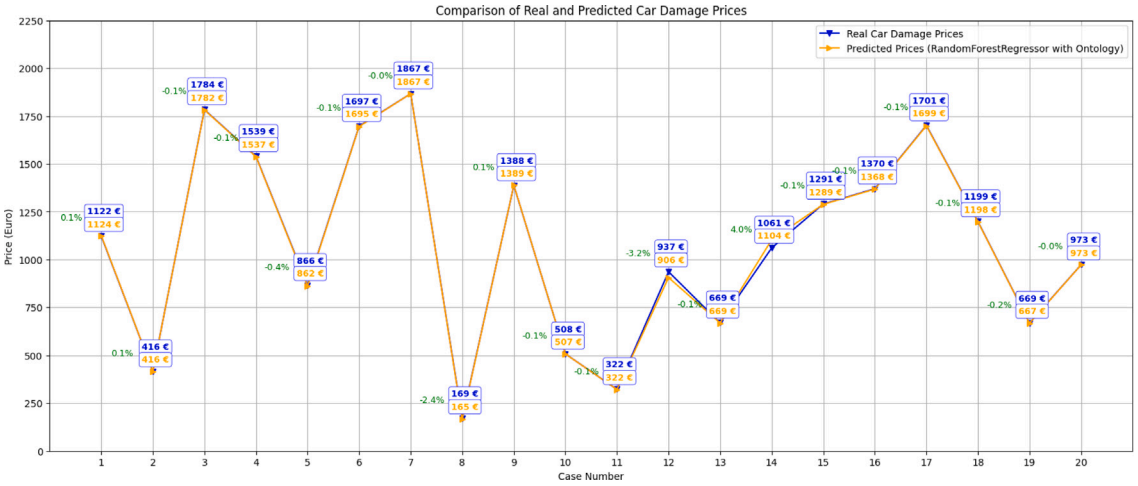

Fig. 9. Comparison of actual and predicted car damage costs for 20 cases using Random Forest in conjunction with ontology cooperation.

efficiently, this article introduces an innovative methodology that integrates regression models with ontology reasoning to significantly improve the accuracy of car damage repair cost predictions. Our experimental evaluation, grounded in a variety of metrics, demonstrates that the inclusion of ontology substantially enhances the predictive performance across all examined models. Notably, the *RandomForestRegresion* and *DecisionTreeRegressor* models, when augmented with our ontology, emerged as the most accurate in estimating repair costs.

The ontology's integration proved instrumental in enabling the regression models to identify pertinent features and understand the complex relationships and dependencies among various factors. This structured approach to domain knowledge has facilitated a deeper insight into the intricacies of car damage repair, resulting in a notable increase in prediction accuracy.

The contributions of this work can be summarized as follows: (i) OCD, an ontology for organizing the complex aspects of car damage. This ontology was populated through the use of named entity recognition and relation extraction techniques. (ii) Definition and incorporation of semantic web rules for reasoning and enriching the features that will be used by regression models. (iii) A methodology that combines ontology with regression models to refine the accuracy of predictions for car damage repair costs. (iv) The validity of our approach was rigorously assessed through a comparative analysis of various regression models, with and without the integration of ontology, utilizing a real-world dataset.

Our research highlights the synergistic potential of combining regression models with ontology reasoning to enhance the precision of car damage repair cost predictions. This integration not only boosts the accuracy of predictions but also deepens our understanding of the car damage repair domain, laying a solid foundation for future investigations aimed at expanding the boundaries of this field.

6. Future work

This article provides a solid foundation, illustrating the viability of our proposed method, yet they also highlight significant opportunities for further refinement and expansion to enhance its utility and effectiveness. Central to future research will be the advancement and improvement of the ontology that underpins our methodology. Prospective studies should focus on the creation of domain-specific, detailed ontologies that cover an extensive array of variables related to the assessment of automobile repair costs. This effort would greatly benefit from collaboration with industry experts, comprehensive data collection, and a commitment to continually update the ontology to reflect new developments and knowledge in the field.

Moreover, the incorporation of additional data sources is anticipated to substantially improve the predictive accuracy of our model. Integrating information such as historical repair records, regional pricing differences, and current market trends can provide deeper insights and enhance the precision of cost estimations. The exploration of sophisticated data fusion techniques and augmentation strategies will be crucial for effectively leveraging these additional data sources.

In addition, the scalability of our method and its practical application in real-world contexts require in-depth investigation. Conducting studies with larger datasets to examine the model's computational efficiency and resource demands will be essential. These studies are crucial for assessing the feasibility of our approach for real-world applications.

CRediT authorship contribution statement

Hamid Ahaggach: Conceptualization, Formal analysis, Investigation, Methodology, Visualization, Writing – original draft, Writing – review & editing. **Lylia Abrouk:** Conceptualization, Formal analysis, Funding acquisition, Investigation, Methodology, Project administration, Supervision, Writing – review & editing. **Eric Lebon:** Data curation, Project administration, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The data that has been used is confidential.

Acknowledgments

This work is supported by both the company **Syartec**[7] and the **ANRT** (National Association for Research and Technology)[8].

[7] Syartec website: https://www.syartec.com.
[8] ANRT website: https://www.anrt.asso.fr.

Table 5
Detailed analysis of car damage assessment for 3 examples where actual and estimated repair costs differ.

Case #	Translated Damage Description	Information Extracted	Ontology Reasoning	Predicted Price	Real Price	Diff.
1	Toyota Yaris with heavily scratched left rear door and left front and left rear fender	Toyota Yaris heavily scratched left rear door left front fender left rear fender.	[Toyota, Yaris, door, left rear, heavy,..., scratch, replacement] [Toyota Yaris fender, left front heavy, ..., scratch, replacement] [Toyota Yaris fender, left rear heavy, ..., scratch, replacement]	442€ + 340€+ 342€ = **1124€**	1122€	2€
12	Ford Focus, damage found: - Left door: deep scratches and a small dent - Rear bumper: minor cracks - Left rear tail light: cracked - Front windshield: small crack	Ford Focus Left door scratches small dent Rear bumper minor cracks Left rear tail light cracked Front windshield small crack	[Ford, Focus, Left, door, small, ..., dent, reparation] [Ford, Focus, Rear, bumper, minor, ..., cracks, reparation] [Ford, Focus, Left rear tail, light, ..., cracked, replacement] [Ford, Focus, Front, windshield, small, crack, replacement]	350€ + 150€ + 206€ + 200€ = **906€**	937€	31€
14	Volvo XC60, damage found: - Front and rear bumpers: broken and perforated -Right exterior mirror: broken -Right front and right rear door: deep scratches -Right front and right rear fender: deep scratches -Broken right front fog lights	Volvo XC60 Front rear bumpers broken perforated Right exterior mirror broken Right front right rear door: deep scratches Right front right rear fender: deep scratches broken right front fog lights	[volvo, XC60, front, bumper, ..., broken, replacement] [volvo, XC60, rear, bumper, ..., broken, replacement] [volvo, XC60, right exterior, mirror, ..., broken, replacement] [volvo, xc60, right front, door, deep, scratches, replacement] [volvo, xc60, right rear, door, deep, scratch, replacement] [Volvo, XC60, right front, fender, deep , ..., scratch, replacement] [Volvo, XC60, right rear fender,deep scratches, replacement] [Volvo, XC60, right front, fog lights, ...,Broken, replacement]	161.3€+ 161.30€+ 102.20€+ 190.40€+ 192.80€+ 100.20€+ 97.40€+ 98.40€ = **1104€**	1061€	43€

Appendix. Online resources

The supplementary online material supports the comprehension and reproduction of this study. It includes: an OWL file representing the ontology (*OCD*), and a file containing SPARQL queries, accessible at the GitHub repository (https://github.com/OntologyCarDamage/OCD) and on the industry portal site (http://industryportal.enit.fr/ontologies/OCD). The demo of the application is available online.[9]

References

Adekitan, A. I., Bukola, A., & Kennedy, O. (2018). A data-based investigation of vehicle maintenance cost components using ANN. *Vol. 413*, In *IOP conference series: materials science and engineering*. IOP Publishing, Article 012009.

Ahaggach, H., Abrouk, L., Foufou, S., & Lebon, E. (2023). Predicting car sale time with data analytics and machine learning. In *Product lifecycle management. PLM in transition times: the place of humans and transformative technologies: 19th IFIP WG 5.1 international conference, PLM 2022, Grenoble, France, July 10–13, 2022, revised selected papers* (pp. 399–409). Springer.

Ahaggach, H., Abrouk, L., & Lebon, E. (2024). Information extraction from automotive reports for ontology population. *Applied Ontology*, (19), 1–30.

Asim, M. N., Wasim, M., Khan, M. U. G., Mahmood, W., & Abbasi, H. M. (2018). A survey of ontology learning techniques and applications. *Database*, *2018*, bay101.

Bishop, C. M., & Nasrabadi, N. M. (2006). *Vol. 4, Pattern recognition and machine learning*. Springer.

Bishop, C. M., et al. (1995). *Neural networks for pattern recognition*. Oxford University Press.

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

Cao, Q., Samet, A., Zanni-Merk, C., de Beuvron, F. d., & Reich, C. (2019). An ontology-based approach for failure classification in predictive maintenance using fuzzy C-means and SWRL rules. *Procedia Computer Science*, *159*, 630–639.

Chandra, R., Tiwari, S., Agarwal, S., & Singh, N. (2023). Semantic rule web-based diagnosis and treatment of vector-Borne diseases using SWRL rules. arXiv preprint arXiv:2301.03013.

Chen, T., & Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM sigkdd international conference on knowledge discovery and data mining* (pp. 785–794).

Chen, C., Liu, Y., Sun, X., Di Cairano-Gilfedder, C., & Titmus, S. (2019). Automobile maintenance prediction using deep learning with GIS data. *Procedia CIRP*, *81*, 447–452.

[9] https://drive.google.com/drive/folders/1o0NOBqxj3rxFURFBmuH-C6kjj_d85TOU

Cook, D. O., Kieschnick, R., & McCullough, B. D. (2008). Regression analysis of proportions in finance with self selection. *Journal of Empirical Finance*, *15*(5), 860–867.

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, *20*(3), 273–297.

Department of Consumer Affairs Bureau of Automotive Repair 10949 North Mather Boulevard Rancho Cordova, C. . (2022). DRAFT vehicle-inspection-manual. URL https://www.bar.ca.gov/pdf/workshops/202207-vehicle-safety-inspection/draft-manual.pdf.

El Massari, H., Gherabi, N., Mhammedi, S., Ghandi, H., Bahaj, M., & Naqvi, M. R. (2022). The impact of ontology on the prediction of cardiovascular disease compared to machine learning algorithms.. *International Journal of Online & Biomedical Engineering*, *18*(11).

Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. *Annals of Statistics*, 1189–1232.

Fudholi, D. H., Maneerat, N., Varakulsiripunth, R., & Kato, Y. (2009). Application of Protégé, SWRL and SQWRL in fuzzy ontology-based menu recommendation. In *2009 international symposium on intelligent signal processing and communication systems* (pp. 631–634). IEEE.

Galton, F. (1886). Regression towards mediocrity in hereditary stature. *The Journal of the Anthropological Institute of Great Britain and Ireland*, *15*, 246–263.

Gareth, J., Daniela, W., Trevor, H., & Robert, T. (2013). *An introduction to statistical learning: with applications in R*. Spinger.

Gegic, E., Isakovic, B., Keco, D., Masetic, Z., & Kevric, J. (2019). Car price prediction using machine learning techniques. *TEM Journal*, *8*(1), 113.

Hastie, T., Tibshirani, R., Friedman, J. H., & Friedman, J. H. (2009). *Vol. 2*, *The elements of statistical learning: data mining, inference, and prediction*. Springer.

Haykin, S. (1998). *Neural networks: a comprehensive foundation*. Prentice Hall PTR.

Hazman, M., El-Beltagy, S. R., & Rafea, A. (2011). A survey of ontology learning approaches. *International Journal of Computer Applications*, *22*(9), 36–43.

Hu, M., & Liu, Y. (2020). E-maintenance platform design for public infrastructure maintenance based on IFC ontology and semantic web services. *Concurrency Computations: Practice and Experience*, *32*(6), Article e5204.

Huang, Y., Huang, K., & Wu, J. (2016). Cost estimation for the pure electric family car's whole life cycle based on partial least square regression. In *2016 4th international conference on electrical & electronics engineering and computer science* (pp. 403–409). Atlantis Press.

Huang, Z., Xu, W., & Yu, K. (2015). Bidirectional LSTM-CRF models for sequence tagging. arXiv preprint arXiv:1508.01991.

Inspektlabs (2023). Automate inspections with AI. URL https://inspektlabs.com.

Jabardi, M. H., & Hadi, A. S. (2021a). Ontology meter for Twitter fake accounts detection. *International Journal of Intelligent Engineering and Systems*, *14*(1), 410–419.

Jabardi, M. H., & Hadi, A. S. (2021b). Using machine learning to inductively learn semantic rules. *Vol. 1804*, In *Journal of physics: conference series*. IOP Publishing, Article 012099.

Jung, S., Pyeon, J.-H., Lee, H.-S., Park, M., Yoon, I., & Rho, J. (2020). Construction cost estimation using a case-based reasoning hybrid genetic algorithm based on local search method. *Sustainability*, *12*(19), 7920.

Kim, J.-M., Yum, S.-G., Park, H., & Bae, J. (2021). A deep learning algorithm-driven approach to predicting repair costs associated with natural disaster indicators: The case of accommodation facilities. *Journal of Building Engineering*, *42*, Article 103098.

Kopitar, L., Kocbek, P., Cilar, L., Sheikh, A., & Stiglic, G. (2020). Early detection of type 2 diabetes mellitus using machine learning-based prediction models. *Scientific Reports*, *10*(1), 11981.

Kyu, P. M., & Woraratpanya, K. (2020). Car damage detection and classification. In *Proceedings of the 11th international conference on advances in information technology* (pp. 1–6).

Lafferty, J., McCallum, A., & Pereira, F. C. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data.

Le, H., Vial, L., Frej, J., Segonne, V., Coavoux, M., Lecouteux, B., et al. (2020). FlauBERT: Unsupervised language model pre-training for French. In *Proceedings of the twelfth language resources and evaluation conference* (pp. 2479–2490). Marseille, France: European Language Resources Association, URL https://aclanthology.org/2020.lrec-1.302.

Lee, S.-K., Kim, K.-R., & Yu, J.-H. (2014). BIM and ontology-based approach for building cost estimation. *Automation in Construction*, *41*, 96–105.

Lessmann, S., & Voß, S. (2017). Car resale price forecasting: The impact of regression method, private information, and heterogeneity on forecast accuracy. *International Journal of Forecasting*, *33*(4), 864–877.

Lipatov, A., Belyanova, E., & Petunina, I. (2024). A multiple linear regression model to predict the biodegradation rate of soil contaminated with different oil concentrations. *Results in Nonlinear Analysis*, *7*(1), 24–34.

Liu, X., Li, Z., & Jiang, S. (2016). Ontology-based representation and reasoning in building construction cost estimation in China. *Future Internet*, *8*(3), 39.

Lu, Y. (2017). Industry 4.0: A survey on technologies, applications and open research issues. *Journal of Industrial Information Integration*, *6*, 1–10.

Martis, J. E., Sannidhan, M., Aravinda, C., & Balasubramani, R. (2023). Car damage assessment recommendation system using neural networks. *Materials Today: Proceedings*.

Massari, H. E., Sabouri, Z., Mhammedi, S., & Gherabi, N. (2012). Diabetes prediction using machine learning algorithms and ontology. arXiv preprint arXiv:1205.5921.

Niknam, M. (2015). *A semantics-based approach to construction cost estimating* (Ph.D. thesis), Marquette University.

Pérez, J., Arenas, M., & Gutierrez, C. (2009). Semantics and complexity of SPARQL. *ACM Transactions on Database Systems*, *34*(3), 1–45.

Puripunyavanich, V., Myojo, S., & Kanazawa, Y. (2005). Estimating the maintenance and repair cost in Life Cycle Cost calculation: A case of automobile ownership in the US. *The Journal of Management Accounting, Japan*, *13*(1–2), 3–23.

Qaddour, J., & Siddiqa, S. A. (2023). Automatic damaged vehicle estimator using enhanced deep learning algorithm. *Intelligent Systems with Applications*, *18*, Article 200192.

Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning*, *1*, 81–106.

Raad, J., & Cruz, C. (2015). A survey on ontology evaluation methods. In *Proceedings of the international conference on knowledge engineering and ontology development, part of the 7th international joint conference on knowledge discovery, knowledge engineering and knowledge management* (pp. 179–186).

Ractable (2023). The speed and accuracy of AI. Now applied to visual assessment. URL https://tractable.ai.

Ruder, S. (2016). An overview of gradient descent optimization algorithms. arXiv preprint arXiv:1609.04747.

Sharma, A., Verma, A., & Gupta, D. (2019). Preventing car damage using CNN and computer vision. *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, *9*, 1–5.

Shearer, R. D., Motik, B., & Horrocks, I. (2008). Hermit: A highly-efficient OWL reasoner. *Vol. 432*, In *Owled* (p. 91).

Smola, A. J., & Schölkopf, B. (2004). A tutorial on support vector regression. *Statistics and Computing*, *14*, 199–222.

Stojadinovic, Z., Kovacevic, M., Marinkovic, D., & Stojadinovic, B. (2017). Data-driven housing damage and repair cost prediction framework based on the 2010 Kraljevo earthquake data. In *Proceedings of the 16th world conference on earthquake engineering, santiago, Chile* (pp. 9–13).

Stone, K., Zwiggelaar, R., Jones, P., & Mac Parthaláin, N. (2022). A systematic review of the prediction of hospital length of stay: Towards a unified framework. *PLOS Digital Health*, *1*(4), Article e0000017.

Tang, X.-B., Liu, G.-C., Yang, J., & Wei, W. (2018). Knowledge-based financial statement fraud detection system: based on an ontology and a decision tree. *Knowledge Organization*, *45*(3), 205–219.

Tchek, A. (2023). Simplify the inspection and remarketing of your vehicles. URL https://www.tchek.ai.

Thirugnanam, M., Thirugnanam, T., & Mangayarkarasi, R. (2013). An ontology-based system for predicting disease using SWRL rules. *International Journal of Computer Science and Business Informatics*, *7*(1).

Tiwari, S., Chandra, R., & Agarwal, S. (2022). Forecasting COVID-19 cases using statistical models and ontology-based semantic modelling: A real time data analytics approach. arXiv preprint arXiv:2206.02795.

Tsarkov, D., & Horrocks, I. (2006). FaCT++ description logic reasoner: System description. In *Automated reasoning: third international joint conference, IJCAR 2006, Seattle, WA, USA, August 17-20, 2006. Proceedings. Vol. 3* (pp. 292–297). Springer.

Xu, M., Zhang, P., Cui, C., & Zhao, J. (2022). An ontology-based holistic and probabilistic framework for seismic risk assessment of buildings. *Buildings*, *12*(9), 1391.

Ye, T., & Liu, B. (2022). Uncertain significance test for regression coefficients with application to regional economic analysis. *Communications in Statistics. Theory and Methods*, 1–18.

Zhang, W., Cheng, Y., Guo, X., Guo, Q., Wang, J., Wang, Q., et al. (2020). Automatic car damage assessment system: Reading and understanding videos as professional insurance inspectors. *Vol. 34*, In *Proceedings of the AAAI conference on artificial intelligence* (pp. 13646–13647).

Zhu, Q., Liu, Y., Shen, Y., & Zhao, Z. (2021). Research on intelligent damage assessment system for time-sharing rental vehicles based on image recognition. *Vol. 1880*, In *Journal of physics: conference series*. IOP Publishing, Article 012012.