Predicting Reasoner Performance on ABox Intensive OWL 2 EL Ontologies[1]

Jeff Z. Pan[1,*], Carlos Bobed[2], Isa Guclu[1], Fernando Bobillo[2], Martin J. Kollingbaum[1], Eduardo Mena[2], and Yuan-Fang Li[3]

[1]University of Aberdeen, Department of Computing Science, Aberdeen, AB24 3UE, U.K.
[2]University of Zaragoza, Aragon Institute of Engineering Research (I3A), Zaragoza, 50018, Spain
[3]Monash University, Faculty of Information Technology, Clayton, VIC 3800, Australia
[*]jeff.z.pan@abdn.ac.uk

ABSTRACT

Reasoner performance prediction for ontologies in the OWL 2 language has been studied so far from different dimensions. One key aspect of these studies has been the prediction of how much time a particular reasoning task for a given ontology will consume. Several approaches have adopted machine-learning techniques to predict time consumption of different reasoning tasks depending on features of the input ontologies. However, these studies have focused on capturing general aspects of the ontologies (i.e., mainly the complexity of their TBoxes), while paying little attention to ABox details. ABox information is particularly important in real-world scenarios, where data volumes are much larger than data-describing schema information. In this paper, we introduce the notion of ABox intensity in the context of predicting reasoner performance and to improve the representativeness of ontology metrics by developing new metrics that focus on ABox features of OWL 2 EL ontologies. Our experiments show that taking into account the intensity through our proposed metrics contributes to overall prediction accuracy for ABox intensive ontologies.

INTRODUCTION

The language OWL 2 DL (Cuenca-Grau et al. (2008)), the most expressive profile of OWL 2, has a worst-case complexity that is 2NEXPTIME-complete (Kazakov (2008)), which constitutes a bottleneck for performance critical applications. Empirical studies show that even the EL profile, with PTIME-complete complexity and less expressiveness, can become too time-consuming (Dentler et al. (2011), Kang et al. (2012b)).

There have been several studies regarding performance prediction of ontologies. Kang et al. (2012a) investigated the hardness category (categories according to reasoning time) for reasoner-ontology pairs and used machine-learning techniques to make a prediction. Using the reasoners FaCT++ (Tsarkov & Horrocks (2006)), HermiT (Glimm et al. (2014)), Pellet (Sirin et al. (2007)), and TrOWL (Pan et al. (2016, 2012), Ren et al. (2010), Thomas et al. (2010)), their prediction had high accuracy in terms of hardness category, but not in terms of reasoning time. In a subsequent study, Kang et al. (2014)

[1] This paper is an extended version of our previous work. Particularly, the work presented here is based on our JIST2016 paper (Guclu, Bobed, Pan, Kollingbaum & Li (2016)) but revised and extended with new metrics to increase the prediction accuracy of the approach.

investigated regression techniques to predict reasoning time. They made experiments, based on their syntactic metrics, using the reasoners FaCT++, HermiT, JFact, MORe (Armas-Romero et al. (2012)), Pellet, and TrOWL. These metrics are generally effective when there is a balance between TBox axioms and ABox axioms. However, our preliminary experiments in Guclu, Bobed, Pan, Kollingbaum & Li (2016) showed that the accuracy of these metrics decreases when the relative size of the ABox with respect to the TBox increases.

We regard this observation important as there are many real-world scenarios where the amount of data exceeds by far the size of the schema associated with them (e.g., Linked Data repositories (Bizer et al. (2009))). Besides, as observed in Yus & Pappachan (2015), there is an increasing interest in using semantic technologies on mobile devices (Bobed et al. (2015)). Given that the ABox constitutes the data of an ontology (Fokoue et al. (2012), Hogan et al. (2011), Ren et al. (2012)), whereas TBox constitutes the schema, on mobile devices, with their restricted resources, TBox axioms are expected to be rather static, whereas the ABox axioms (data) tend to change more frequently. Thus, due to volume and dynamism, an approach that can capture the influence of the ABox in reasoning performance in a more accurate way is needed to make accurate overall predictions. Plenty of applications can benefit from this prediction mechanism, both in resource-limited scenarios as well as in non-limited ones. For example, on the one hand, having an accurate processing time prediction can be combined with battery consumption prediction (Guclu, Li, Pan & Kollingbaum (2016)) to devise new adaptive methods for reasoning in mobile devices. On the other hand, semantic applications dealing with highly volatile data can also benefit from these predictions to decide whether or not to update the materialization of their knowledge (Bobed et al. (2014)).

In this paper, we aim to investigate which metrics could help to further improve reasoner performance predictions in the presence of ABoxes that are significantly different in size than the TBoxes. Thus, we propose a framework to devise ontology metrics where the estimated complexity of the TBox is propagated to the ABox. First of all, we introduce the notion of ABox intensity, which is defined as the ratio between the size of the TBox and the ABox of an ontology, and we use it to determine so-called ABox intensive ontologies, i.e., those ontologies whose ABox intensity is above a domain dependent threshold (in our particular experiments, we set such a ratio threshold to 5).

Our main contributions can be summarized as follows:

- We introduce the notion of ABox Intensity to be taken into account in the prediction and analysis of ontology reasoning performance.
- We propose to extend the previously available metrics proposed by Kang et al. (2014) with a set of metrics (51) that are designed to: 1) capture the complexity introduced by the ABox Intensity of the ontology, and 2) capture the combined structural complexity of TBox and ABox. In this work, structural complexity means a numerical value that tries to estimate the influence of structures of some given TBox and ABox on reasoning time.
- We show that our proposed new metrics increase the accuracy in predicting time consumption of ABox intensive ontology reasoning. Besides, we also validate their contribution by applying a feature selection algorithm, which express that our metrics are effectively selected in these scenarios.

The rest of the paper is organised as follows. In the next section, we present the background knowledge for our work. Then, we present some related works to contextualize our proposal, and explain our research objectives with the core motivation of this research. Next, the newly proposed metrics are detailed. We continue by outlining experimental settings and presenting some results. Finally, we draw conclusions and outline future work.

BACKGROUND KNOWLEDGE

In this section, we will briefly introduce basics about ontology reasoning. Our work is focussed on reasoning over OWL 2 EL ontologies, both for processing terminological closure (TBox) and for full materialization (as it considers both TBox and ABox).

An ontology consists of a set of axioms that are statements describing (1) relations between class (property) descriptions, (2) characteristics of properties, such as asserting that a property is transitive, or (3) instance-of relations between individuals and classes, or between pairs of individuals and properties, as described by Pan (2004). For example, an axiom can be of the following form:

DisjointClasses(:Animal : Plant) (1)

It can be interpreted (Cuenca-Grau et al. (2008)) as "Nothing can be both an :Animal and a :Plant". These axioms encode knowledge about the concepts (classes) mentioned above – we can state that an ontology comprises knowledge or represents a "knowledge base".

Ontologies expressed in Description Logic (Baader et al. (2003)) are comprised of two parts: the TBox and the ABox. Whereas the TBox provides the "terminological component" of the ontology, the ABox constitutes the "assertion component" – facts associated with concepts in this knowledge base. Within the set of TBox axioms, we want to highlight *General Concept Inclusion* axioms (GCIs), and *Role Inclusion* axioms (RIAs): A GCI axiom states that a concept C_1 is a subclass of another concept C_2 or, in other words, that C_2 subsumes C_1. Similarly, a RIA axiom encodes the fact that a chain of properties $OP_1..OP_n$ is a subproperty of another property OP_j.

In our study, we have chosen the OWL 2 EL profile due to its polynomial-time complexity for basic reasoning problems. This complexity characteristics proves advantageous in applications that are dealing with ontologies containing very large numbers of properties and/or classes, as recommended by W3C (2009). The supported concepts in OWL 2 EL are atomic A, conjunction $C_1 \sqcap C_2$, (concrete and abstract) existential restriction $\exists OP.C$ and $\exists DP.\boldsymbol{d}$, value restriction $\exists OP.\{a\}$, singleton nominal $\{a\}$, and local reflexivity $\exists OP.self$, where *DP* is a datatype property and **d** is a data range. In OWL 2 EL, it is common to distinguish some specific types of GCIs and RIAs that are commonly used in practice, namely disjoint concepts $Disj(CE_1, CE_2)$, domain *Dom(OP, CE)* or *Dom(DP, **d**)*, range *Rng(OP, CE)* or *Rng(DP, **d**)*, reflexivity *ref (OP)*, transitivity *trans(OP)*, and, only in the case of data properties, functionality *funct(DP)*. Further characteristics of the EL profile can be analysed online (W3C (2009)).

Finally, we introduce some reasoning tasks which are important for our study:

- **Classification** of an ontology: This reasoning task consists of computing a hierarchy of concepts

(resp. roles) based on their subsumption relations, that is, by deciding for every pair of atomic concepts (resp. atomic properties) A_1 , A_2 whether A_1 is a subclass (resp. subproperty) of A_2 or not.

- **Materialization**: This reasoning task consists of computing all entailed instances of every atomic concept over both TBox and ABox. As a result, the performance of full materialization tasks is affected by the features describing the TBox aspect and ABox aspect of the ontology.

ABox Intensity According to our recent research (Guclu, Bobed, Pan, Kollingbaum & Li (2016)), an important dimension of ontologies has not been analysed yet, i.e., the *intensity* of a set of ABox axioms. We define *ABox intensity* of ontology as the ratio of the count of ABox axioms to TBox axioms. Accordingly, we define ontology as being *ABox intensive* when its ABox intensity is above a domain-dependent threshold. In this paper, we are going to define ontology as ABox intensive when it has ABox intensity above 5.0[2]. Bear in mind that we do not claim a particular fixed value (5, 10, etc.) as a right/optimum intensity ratio. However, we believe that different ABox intensities with different profiles and contexts will show different behaviours that deserve to be investigated. As observed by Hu et al. (2011), ontologies from different domains can have different features that can cause different behaviours in terms of performance. In this paper we question whether ontology sets with different ABox intensities show the *same* behaviour. We assume that the dimension of *ABox intensity* of ontology is as important as other crucial features, such as the domain and the profile. Disregarding this dimension may produce misleading results and wrong conclusions about complexity issues in reasoning.

Ontology Size Hu et al. (2008) considers an ontology as *large*, if it contains more than 1000 entities, and proposes an approach how to efficiently process them. In the ORE 2013[3] Workshop, ontologies were categorized according to their size as *small* ((0-499]), *medium* ([500-4999]), *large* ([5000-∞) by counting the logical axioms in the original ontology (that is, before doing any reasoning) (Gonçalves et al. (2013)). We will follow the ontology categorizing methods according to their size proposed in ORE 2013.

RELATED WORK AND TECHNICAL MOTIVATION

Ontology metrics have been developed to capture particular features of ontologies that impact on the complexity of ontology reasoning, such as cohesion (Yao et al. (2005)), quality (Burton-Jones et al. (2005)), or population task (Maynard et al. (2006)). These metrics have been used to analyse ontology reasoning in terms of complexity by Zhang et al. (2010), and energy consumption on mobile devices by Guclu, Li, Pan & Kollingbaum (2016).

Kang et al. (2012a) proposed a set of metrics to classify raw reasoning times of ontologies into five large categories: [0s.–100ms.], (100ms.–1s.], (1s.–10s.], (10s.–100s.] and (100s.–∞). Despite a high accuracy of prediction of over 80%, this approach does not provide actual reasoning time, but time categories (which might need to be adapted for different scenarios and, therefore, might require to retrain the model). However, predicting actual reasoning times may be essential for particular systems and scenarios.

In 2014, Kang et al. (2014) extended their work and proposed a new set of metrics to predict actual reasoning time by developing regression models. They extended the previous 27 metrics developed by

[2] In our previous work (Guclu, Bobed, Pan, Kollingbaum & Li (2016)), we had generated a dataset with an ABox intensity of 10.
[3] http://curation.cs.manchester.ac.uk/ore2013

Kang et al. (2012a) and Zhang et al. (2010) and developed a set of 92 metrics that include 24 ontology-level (ONT) metrics, 15 class-level (CLS) metrics, 22 anonymous class expression (ACE) metrics, 30 property definition and axiom (PRO) metrics, and ontology size[4].

While a high number of metrics are usually proposed by researchers, Sazonau et al. (2014) proposed instead a local method that involves selecting a *suitable* small subset of the ontology and use extrapolation to predict total time consumption of ontology reasoning using the data generated by the processing of such a small subset. To do so, they used *Principal Component Analysis* (PCA) (Jolliffe (2002)). In their experiments, Sazonau et al. (2014) observed that 57 of the studied features could be replaced by just one or two features. Using a sample size of 10% of the ontology for reasoning, they argue that they reached good predictions with simple extrapolations. They list advantages of their method as: 1) more accurate performance predictions, 2) not relying on an ontology corpus, 3) not being biased by this corpus, and 4) being able to obtain information about a reasoner 's behaviour of resource consumption using such a small set of ontologies. A remarkable contribution of this approach is that it reduces the difficulty of selecting an unbiased corpus (Matentzoglu et al. (2013)), which is needed for checking the validity of the prediction model and the accuracy of the prediction. However, predicting reasoning with 10% of ontology may not always be applicable, especially when the ontology requires high reasoning times.

Technical Motivation As denoted by Della Valle et al. (2013), semantic processing of massive sets of complex and highly dynamic data necessitates performance metrics and a systematic roadmap about how to process this massive and dynamic data. Furthermore, many smart applications, such as those that process data sets captured by sensors and that are growing fast in terms of size, mainly have to deal with ABox information. The TBox of ontologies tends not to change as frequently as the ABox (Bobed et al. (2014)). This fact necessitates applications to be able to manage the changes in an ABox and be able to predict the performance of ABox reasoning accordingly.

Urbani et al. (2011) observed in their experiment, which compared the computational cost of reasoning just with the TBox with that of complete ontological closure (TBox and ABox), that computing the full closure is 1–2 orders "larger" than computing just the TBox (see Table 1). In this experiment, they processed two real (LLD[5] , LDSR[6]), and one artificial (LUBM (Guo et al. (2005))) ontologies on WebPIE (Urbani et al. (2010)). The computational cost of processing the ABox, in addition to the TBox, leads us to think that the ABox constitutes the main challenging and resource consuming part (Urbani et al. (2011)). Besides, we have to take into account that the **real size** of the terminological knowledge (i.e., the number of TBox axioms) can be huge with respect to the size of the factual knowledge (i.e., the number of ABox axioms), as pointed out by van Harmelen (2011).

To see whether available metrics can be used to predict time consumption of ontology reasoning, we implemented the 92 metrics proposed by Kang et al. (2014), and ran their experiments using the 1941 *OWL 2 EL* ontologies in the ORE 2014 dataset, instead of the 451 real-world ontologies that were used in the original experiments. The result was interesting insofar as the coefficient of determination R^2 decreased sharply from 93.40% to 61.45%, which can be seen in Figure 1. According to our experiments

[4] While Guclu, Bobed, Pan, Kollingbaum & Li (2016) and Kang et al. (2014) do not include ontology size as one of the 91 metrics, actually they also use such a parameter in their experiments, so they consider a total of 92 metrics.
[5] LinkedLifeData, available at http://linkedlifedata.com/
[6] Also known as FactForge, available at http://factforge.net/

(detailed in *Results and Evaluation* Section), available metrics capture the complexity of the ontologies to some extent, mainly the TBox complexity aspect, and are appropriate for ontologies with non-intensive ABoxes. However, when ABox/TBox ratio increases, which is the inevitable real-world situation, available metrics start to lose their accuracy when it comes to predicting the time consumption of ontology reasoning.

	Classification		Materialization	
Input	Time (sec.)	# axioms	Time (sec.)	# axioms
LDSR (862M)	89	0.62M	10036	927M
LLD (694M)	332	7.06M	3931	330M
LUBM (1101M)	8	22	4526	495M

Table 1. Comparison of classification against materialization

Currently, there is no general approach for predicting how reasoners will perform with ontologies of arbitrary characteristics, such as size, ABox / TBox ratio, context, etc. However, in this paper we make a first step towards it by proposing a new approach for predicting resource requirements of ontology reasoning. In particular, we propose a detailed analysis of ontology characteristics that provides a deeper insight into the nature of ontologies, and its impact on reasoning performance and resource requirements. Our aim is to increase the predictability of ontology reasoning performance by developing metrics that will increase the accuracy of prediction in the presence of high ABox/TBox ratios. We believe that this research will support a more feasible implementation environment for semantic technologies.

EXTENDING THE ONTOLOGY METRICS SET

As mentioned above, our research investigates ABox intensive ontologies, which we define as those whose ratio of ABox/TBox axioms is above a given threshold (in our current work, we have set it to 5). Some of the 92 metrics proposed by Kang et al. (2014) are obtained by transforming ontology into a graph in order to capture the relationship between ABox and TBox axioms. However, their approach calculates the effect of ABox axioms only up to a certain extent. It is apparent that connected ABox axioms potentially cause more inferences than disconnected ABox axioms. These connections can increase the reasoning time substantially if the TBox is complex. This is coherent with the results obtained in our previous work (Guclu, Bobed, Pan, Kollingbaum & Li (2016)), where we already observed that the models trained with this set of 92 metrics began to lose accuracy in predicting time consumption of ontology reasoning when the ABox/TBox ratio increased.

Thus, apart from using the already 92 proposed metrics, we propose to include the propagation of the complexity of the TBox into the ABox, and to treat each of the instance axioms in the ABox as witnesses of such complexities in the ontology. For this purpose, we started with extending this set of metrics with our 15 *Class Complexity Assertions* (CCA) metrics in Guclu, Bobed, Pan, Kollingbaum & Li (2016), which contributed to performance prediction of ontologies that are ABox intensive (i.e., they exhibit a high ABox/TBox ratio). In this current work, we have revisited the definition of CCA metrics to include the complexity of the involved roles and datatype properties, as well as to add the effects of the General Concept Inclusions (GCIs). The result is five sets of metrics: *Intensity Metrics* (IM), *Concept Complexity*

Assertions with GCIs applied (CCA')[7], *Concept Complexity Assertions* without GCIs applied (CCA_WO), *Object Property Complexity Assertions* (OPCA), and *Datatype Property Complexity Assertions* (DPCA).[8]

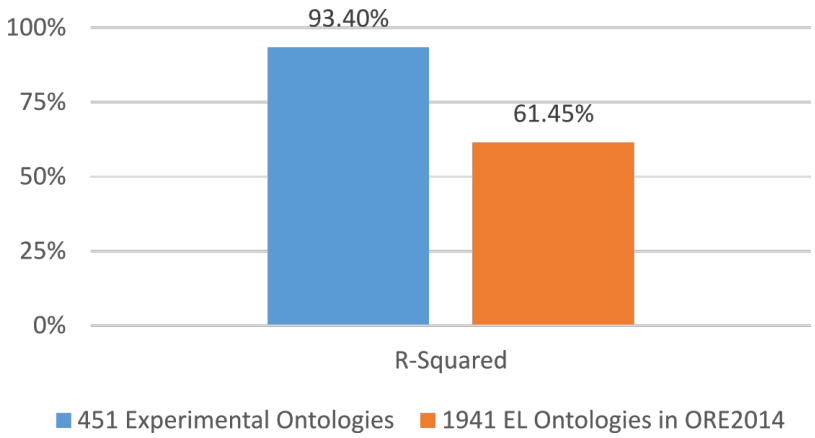

Figure 1. Comparison of R^2 values between 451 ontologies and ORE 2014 dataset

The first set (IM) is composed by the following metrics:

- *TBoxSize*: The number of TBox axioms obtained from OWLAPI.
- *ABoxSize*: The number of ABox axioms obtained from OWLAPI.
- *ABoxTBoxRatio*: The ratio of ABox axioms to TBox axioms.

For each of the rest of the sets of metrics (CCA', CCA_WO, OPCA, and DPCA), we can distinguish two different subsets: the inner complexity values, and the witnessed complexities. In brief, the first set is an aggregated estimation of the complexity of each of the considered ontology elements (i.e., concept expressions, object properties, and datatype properties); the second one is obtained by considering each instance axiom (i.e., class or role assertion) as a witness of the associated ontology element, and aggregating the weighted values.

In the rest of the section, we firstly present how the estimations of the complexity of each of the single considered ontology elements and their number of witnesses are obtained, and then, we move onto how these values are aggregated to obtain the final sets of metrics for each type of ontology elements.

Complexity Estimation of the Considered Ontology Elements

First of all, to calculate the metrics, we estimate the complexities of the different elements in the ontology. We gather such values following the three steps shown in Figure 2:

1. *Role Complexity Estimations*: We estimate the complexity of the roles (object and datatype properties) in the signature of the ontology. In a second step, we use the RIAs to adjust such

[7] We add the apostrophe in order to avoid confusing them with the ones presented in Guclu, Bobed, Pan, Kollingbaum & Li (2016).

[8] 7The source codes of all the metrics presented in this paper are accessible online at http://sid.cps.unizar.es/projects/OWL2Predictions/IJSWIS17/

complexity values.

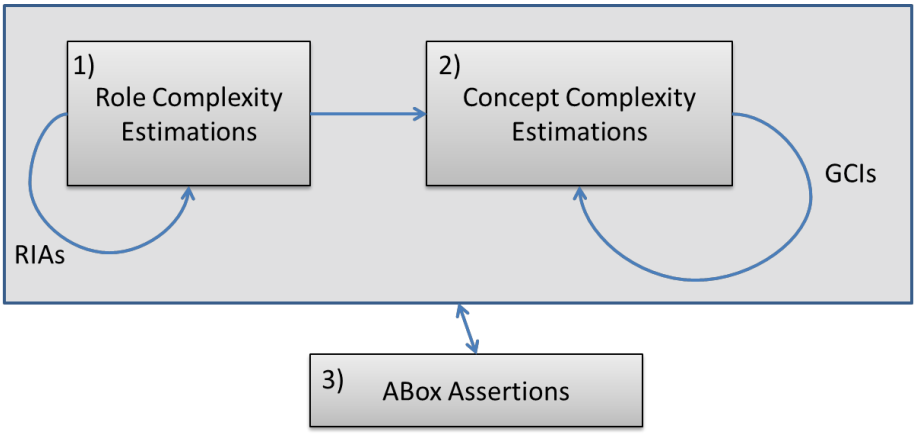

Figure 2. Information taken into account and steps performed to calculate the metrics.

2. *Concept Complexity Estimations*: We gather all the concept expressions that are present in the ontology, and build an initial table with the inner complexity estimations. This table is built taking into account the estimated role complexities. Using these initial complexity values, we apply the GCIs (all the expressions in the GCIs have been previously gathered) to adjust the actual estimated complexity. As we will see, this is done in a non-reentrant way (i.e., all the GCIs affecting a concept expression are considered at once to avoid having to recalculate them until they converge). As a result, we have an adjusted estimation of all the concept complexities of the concept expressions appearing in the axioms of the ontology.
3. *ABox Assertions*: Finally, we use the ABox assertions to compute the witnesses of the estimated complexity captured in the previous tables.

The values of the different metrics will be obtained from the estimated complexity values of the different elements and the counts of witnesses. The rationale behind all the estimations of the different elements is to take into account the number of individuals/assertions that each of them is going to introduce in the ABox materialized graph. In the following, we detail the estimation of the complexity of the different elements, presented in the same order as they are calculated.

OPCA and DPCA Metrics - Complexity Estimation

For each of the object properties OP_i in the signature of the ontology O, we compute the inner complexity as follows:

$$comp(OP_i) = innerComp(OP_i) + RIAsComp(OP_i)$$

where:

$$innerComp(OP_i) = 1.0 + trans(OP_i) + ref(OP_i)$$

with

$$trans(OP_i) = \begin{cases} 2.0 \; if \; OP_i \; is \; transitive \\ 0.0 \; otherwise, \end{cases}$$

$$ref(OP_i) = \begin{cases} 1.0 \; if \; OP_i \; is \; reflexive \\ 0.0 \; otherwise, \end{cases}$$

and

$$RIAsComp(OP_i) = \; subPropComp(OP_i) + subChainPropComp(OP_i)$$

with

$$subPropComp(OP_i) = \; \left|\{OP_i \sqsubseteq OP_j \mid OP_i \neq OP_j\}\right|$$

$$subChainPropComp(OP_i) = \; \left|\{OP_1..OP_n \sqsubseteq OP_j \mid i \in \{1..n\} \wedge (\neg Trans)\}\right| + 2 * \left|\{OP_1..OP_n \sqsubseteq OP_i | Trans\}\right|$$

where $Trans$ evaluates to true if the subChainProperty axiom $OP_1..OP_n \sqsubseteq OP_j$ codifies transitivity.

Similarly, for each of the datatype properties DP_i in the signature of the ontology O, we compute their inner complexity as follows:

$$comp(DP_i) = \; innerComp(DP_i) + subDataPropComp(DP_i)$$

where:

$$innerComp(DP_i) = \; 1.0 + func(DP_i)$$

with

$$funct(DP_i) = \begin{cases} 1.0 \; if \; DP_i \; is \; functional \\ 0.0 \; otherwise, \end{cases}$$

and:

$$subDataComp(DP_i) = \left|\{DP_i \sqsubseteq DP_j \mid DP_i \neq DP_j\}\right|$$

Note that the object properties cannot be transitive and functional at the same time due to decidability problems (Cuenca-Grau et al. (2008)), and datatype properties, by definition, cannot be transitive.

CCA' and CCA WO Metrics - Complexity Estimation

Once we have calculated the inner complexity of all the object and datatype properties in the signature of the ontology, we can estimate the complexity of the concept expressions. The CCA_WO and CCA' sets of metrics just differ in whether GCIs are applied or not to the concept expression estimation, so in the following, we will focus on explaining the CCA' calculation.

Thus, let be $N_{CE} = \{CE_i | \tau \in O\}$ where $\tau \; \epsilon \; \{a: CE_i, CE_1 \sqsubseteq CE_2, Disj(CE_1, CE_2), Dom(R, CE_i), Rng(R, CE_i)\}$ is a logical axiom of the ontology O. For each $CE_i \in N_{CE}$, we estimate its complexity as follows:

$$comp(CE_i) = \frac{height(CE_i) + sigSize(CE_i) + cost(CE_i)}{3}$$

With:

- $height(CE_i)$ being the height of the expression as a parsing tree. In Table 2, its recursive definition can be found ($Height$ column).
- $sigSize(CE_i)$ being the number of different ontology terms (i.e., atomic concept names, object and datatype properties, and instances) that appear in the expression.
- $cost(CE_i)$ being the estimated cost of the concept expression taking into account the different constructors. It is calculated by recursively applying the costs presented in Table 2 ($Cost$ column).

Concept Expression Atom	Cost	Height
A	1	1
$C \sqcap D$	$cost(C) + cost(D)$	$max\big(height(C), height(D)\big) + 1$
$\exists R.C$	$cost(R) + cost(C)$	$height(C) + 1$
$\exists R.\{a\}$	$cost(R)$	1
$\exists S.self$	$cost(S)$	1
$\{a\}$	1	1
$\exists T.\boldsymbol{d}$	$cost(T)$	1
$\exists R.\{v\}$	$cost(T)$	1

Table 2. Estimated costs of the different basic concept expressions in OWL EL profile.

At this point, we have obtained a table of isolated estimation values (i.e., they do not take into account the possible interactions due to GCIs), which are the values, which will be used to calculate the CCA_WO metrics.

To include GCIs in the estimation, we had to device a way which were independent of the axiom processing order and which did not implied to reason (otherwise, why should we want to predict the cost of the reasoning?)[9]. Thus, to obtain an estimation of the complexity introduced by the GCIs in a deterministic way for the different concept expressions in N_{CE}, we focus only on the axioms where they appear in the left-hand side of the axiom. So, let be

$$GCISuperElems(CE_i) = \{CE_j \mid \{CE_i \sqsubseteq CE_j\} \in O \wedge CE_i \neq CE_j\}$$

and

$$GCISuper(CE_i) = \{CE_j \mid CE_i \in GCISuperElems(CE_i)\}$$

Then, we calculate the final estimated complexity of a given concept expression as:

$$compWGCI(CE_i) = comp(CE_i) + comp(GCISuper(CE_i))$$

The intuition is that, focusing on the ABox interactions, if we assert any individual to belong to *CE$_i$*, the

[9] We could process all the axioms C ⊑ D one by one, propagating the complexity from left to right, but the results would depend on the order that the axioms were processed as we are not working with the materialized taxonomy.

reasoning algorithm will have to assert at least that it also belongs to the concept expressions that explicitly subsume CE_i.

Counting Witnesses

As mentioned above, in order to propagate the complexity of the TBox to the ABox, we consider each of the ABox assertions as a witness of the complexity of the asserted element within the ontology. Thus, apart from estimating the complexity of the elements in the ontology, we count how many ABox assertions affect each of them.

For each of the concept expressions $CE_i \in N_{CE}$, we account their witnesses as:

$$witnessCount(CE_i) = \left|Ind_{CE_i}\right| + \left|DomOP_{CE_i}\right| + \left|RngOP_{CE_i}\right| + \left|DomDP_{CE_i}\right|$$

with

$$Ind_{CE_i} = \{a |\ a \in Ind(O) \wedge\ CE_i(a) \in O\}$$

$$DomOP_{CE_i} = \{R(a,b) |\ R(a,b) \in O\ \wedge\ Dom(R) = CE_i\}$$

$$RngOP_{CE_i} = \{R(a,b) |\ R(a,b) \in O\ \wedge\ Rng(R) = CE_i\}$$

$$DomDP_{CE_i} = \{T(a,v) |\ T(a,v) \in O\ \wedge\ Dom(T) = CE_i\}$$

Note that *Dom(R)* and *Rng(R)* here refer to the domain and ranges *explicitly* asserted (that is, they only are counted if $Domain(R/T\ , CE_i\ )$ or $Range(R, CE_i\ )$ axioms are included in the ontology O.

For the witnessed complexity of object and datatype properties, we use the cardinality of

$$RA_{OP_i} = \{OP_i\ (a,b) | a \in Ind(O) \wedge b \in Ind(O) \wedge OP_i\ (a,b) \in O\}$$

for the object properties, and the cardinality of

$$DA_{OP_i} = \{DP_i\ (a,v) | a \in Ind(O) \wedge DP_i\ (a,v) \in O\}$$

for the datatype properties, respectively.

Finally, we apply a Laplace smoothing[10] to include also into the metrics the ontology elements which appear in the ontology signature but do not have any explicit individual assertion.

Computing the Metrics

Once we have all the estimated values and the witnesses for each concept expression and role (object and datatype properties) in the ontology, we aggregate their values to obtain the final values of the metrics. Firstly, for each of the different sets of estimations, we calculate its total sum, average value, maximum and minimum values, standard deviation, and entropy of the complexity distribution. Secondly, we

[10] Taken from Natural Language Processing, basically, it consists in adding 1 to all the witnessed values of the considered ontology elements in the ontology.

introduce the witnesses into the equations, and we obtain the same aggregated values, but weighted using the witnesses counts of each considered ontology element.

For illustrative purposes, let us consider CCA' metrics. Thus, firstly, we would obtain a set of inner complexity metrics:

- $TCCA'$: Total amount of estimated complexity of the ontology O (i.e., the concept expressions in N_{CE}).

$$TCCA' = \sum_{CE_i \in N_{CE}} compWGCI(CE_i)$$

- AVG_CCA': Mean estimated complexity of the concept expressions in N_{CE}.

$$AVG\_CCA' = \frac{TCCA'}{|N_{CE}|}$$

- MAX_CCA': Maximum estimated complexity of the concept expressions in N_{CE}.
- MIN_CCA': Minimum estimated complexity of the concept expressions in N_{CE}.
- STD_CCA': Standard deviation of complexity of the concept expressions in N_{CE}.
- ENT_CCA': Entropy of the complexity distribution of the concept expressions in N_{CE}.

$$ENT\_CCA' = \sum_{CE_i \in N_{CE}} \left(\frac{comp(CE_i)}{\sum_{CE_i \in N_{CE}} comp(CE_j)} \cdot \log_2 \left(\frac{comp(CE_i)}{\sum_{CE_i \in N_{CE}} comp(CE_j)} \right) \right)$$

Then, we would obtain a set of witnessed metrics:

- $TWCCA'$: Total witnessed complexity of the ABox, which is calculated summing all the products of the estimated complexities of the concept expressions with their *witness individuals*.

$$TWCCA' = \sum_{CE_i \in N_{CE}} compWGCI(CE_i) * witnessCount(CE_i)$$

- AVG_WCCA': Mean witnessed complexity of the ABox of the concept expressions in O.

$$AVG\_WCCA' = \frac{TWCCA'}{|N_{CE}|}$$

- MAX_WCCA': Maximum witnessed complexity of a concept expression in O.
- MIN_WCCA': Minimum witnessed complexity of a concept expression in O.
- STD_WCCA': Standard deviation of witnessed complexity of the concept expressions in O.
- ENT_WCCA': Entropy of the witnessed complexity distribution of the concept expressions in O. It is calculated in a similar way to ENT_CCA', but in this case, the cardinality associated to each concept expression is its *witnessed complexity* (i.e., its estimated complexity multiplied by its count of witnesses).

In the case of CCA_WO, OPCA, and DPCA metrics, we substitute *compWGCI* function by the appropriate comp function. Finally, note that in the case of CCA_WO metrics, the witnesses counts will be the same as for the CCA' metrics, but the estimated complexity values of the concept expressions will

differ as, in this case, the GCIs were not taken into account.

EXPERIMENTAL SETUP

We empirically validated our hypothesis about the *ABox intensity* dimension of ontologies and contribution of new metrics. All our experimental setup, scripts and results are available online[11].

Evaluation metrics for assessing the prediction accuracy of models generated with available (92) metrics and combined (143) metrics are listed in *Evaluation Metrics* Subsection. All steps regarding data collection and techniques used in generating/improving models are explained in *Data Collection and Techniques Used* Subsection.

Evaluation Metrics

R^2 and $MAPE$ are used to decide whether our regression model is valid for describing the relation between our metrics and the predictions made by the model. The coefficient of determination (R^2) is a crucial output of regression analysis, indicating to what extent the dependent variable is predictable. For example, a value 0.91 for R^2 means that 91% percent of the variance in Y is predictable from X. Let $y(t)$ be the observed value of y in second t, $\hat{y}(t)$ be the predicted value for y in second t, and $\bar{y}$ be the mean of the observed values, then:

$$R^2 = \frac{\sum_t (\hat{y}(t) - \bar{y})^2}{\sum_t (y(t) - \bar{y})^2} \quad (2)$$

The *Mean Absolute Percentage Error (MAPE)* is a measure of prediction accuracy of a prediction method in statistics that is used to expresses accuracy as a percentage. For calculating the *MAPE* of our prediction model, we will divide the difference of observed and predicted values, divide this by the observed values, and get the average of all observations in the scope.

$$MAPE = 100 * \frac{\sum_{t=1}^{n} \frac{|\hat{y}(t) - y(t)|}{y(t)}}{n} \quad (3)$$

[11] https://github.com/IsaGuclu/ReasoningABoxIntensiveOntologies

Data Collection and Techniques Used

Reasoners We have used ELK 0.4.3, TrOWL 1.5. and JFaCT 1.2.4 for testing experimental ontologies, and we have selected *ABox Materialization* with all three as our experimental task. Note that materialization starts by classifying the ontology, so TBox reasoning is also performed.

In our experiments, we implemented ABox materialization with one thread (i.e., no parallelization is applied). We are aware that we could benefit from parallelization in ABox materialization, and it would improve the performance (Ren et al. (2012)) to some extent. However, as RAM I/O becomes the bottleneck because of the limited bandwidth (Ren et al. (2012)) of the RAM when many worker threads compete for RAM access and this would cause some side effects in measuring the execution time, we preferred to leave the performance prediction of parallel ABox materialization as future work. Finally, we set a timeout of 30 minutes to each ontology processing to limit the amount of time required to gather all the data.

	Dataset-1	Dataset-2
ELK	1909	3858
JFact	1858	3774
TrOWL	1905	3839

Table 3. Number of correctly processed ontologies in our experiments.

	Small	Medium	Big
0 <= *Ratio* <	308	468	810
5 <= *Ratio* <	62	80	13
10 <= *Ratio*	30	109	61
Total	400	657	884

Table 4. Distribution of ontologies in dataset-1.

Ontologies We have used 2 datasets for training our model:

1. *Dataset-1 (1941 Instantiation OWL 2 EL Ontologies in ORE 2014[12]):* This dataset contains 1941 ontologies in EL instantiation experiment set from 16,555 ontologies in ORE 2014 dataset. This dataset will be abbreviated as "DS1" in figures and tables.
2. *Dataset-2 (3858 ABox-intensive Ontologies obtained via Data Augmentation):* Plentiful high-quality data is a key factor in training machine learning models to expect good prediction accuracy. Unfortunately, *normal* real-world datasets may contain *abnormal* or *interesting* samples that will misguide your models in training and produce unexpected wrong predictions. Thus, to avoid that sort of misleading training scenarios, we preferred augmenting our dataset according to our target scope, i.e. ABox intensive ontologies. Recall that, in this experimental setup, we are

[12] https://zenodo.org/record/10791

defining ontology as *ABox intensive*, if the count of ABox axioms in such ontology is at least 5 times the count of TBox axioms. So, we filtered 356 ABox-intensive ontologies in Dataset-1, and we produced up to 10 new ontologies from each of them using the TBox of the original ontology and randomly selecting subsets of the ABox axioms of the original ontology. The result is our Dataset-2 ("DS2" in figures and tables), which contains 3858 ABox-intensive ontologies. Source code of the data augmentation procedure and an implementation with executable files as used in the experiments are accessible online[13]. When gathering the execution times, we came across with inconsistent ontology exceptions, as well as with ontologies that could not be processed before the timeout. Table 3 shows how much ontology from each dataset each reasoner correctly processes. Tables 4 and 5 show the distribution of ontologies in datasets 1 and 2, respectively.

	Small	Medium	Big
$0 <= Ratio < 5$	0	0	0
$5 <= Ratio <$	976	1033	182
$10 <= Ratio$	392	858	417
Total	1368	1891	599

Table 5. Distribution of ontologies in dataset-2.

Prediction Model Construction The importance of a correct selection of the model and its outcome is widely documented in various areas (Kohavi (1995), Sleeman et al. (1995), Burnham & Anderson (2002), Ozkan (2016)). Inspired by the consistent high accuracy of the Random Forest based regression models in the study in Kang et al. (2014), we adopted the same approach using the metrics as predictor variables. Regular 10-fold cross-validation is performed to ensure the generalizability of the model. To see the validity of the model, we also randomly separated the datasets into 80% training set and 20% test set and measured the evaluation metrics.

Feature Selection Feature selection is an important step in machine learning methods as large feature sets may become inconvenient in terms of: 1) high performance requirements, and 2) decrease in accuracy due to the noise introduced by unrelated features. Besides, in our setting, feature selection is also important as it provides an objective measure of how important metrics are in a prediction model, and we wanted to see how many of our new metrics contributed actively to the model generated by the machine learning algorithm.

In our experiments, we have used the Boruta Algorithm, proposed by Kursa & Rudnicki (2010). It is "based on the same idea that forms the foundation of the random forest classifier, namely, that by adding randomness to the system and collecting results from the ensemble of randomized samples one can reduce the misleading impact of random fluctuations and correlations"(Kursa & Rudnicki (2010)).

[13] https://github.com/IsaGuclu/ReasoningABoxIntensiveOntologies

RESULTS AND EVALUATION

In our study, we investigated the reasoning performance of a reasoner and ontology characteristics represented by the already available metrics (92 metrics by Kang et al. Kang et al. (2014)), and our 51 new metrics (CCA', CCA WO, OPCA, DPCA). While developing our new metrics, we aimed at capturing the complexity of ontologies without losing accuracy when ABox intensity increased. Our goal is to make prediction models that can be applied to ABox intensive ontologies with high stability, using metrics that can represent the complexity of the ontology as accurately as possible.

In order to assess the quality of our metrics, we ran two different sets of experiments evaluating the accuracy of the prediction in two different ways: 1) using 10-fold Cross Validation, and 2) validating them by splitting the datasets into 80%/20% training/test data. This allowed us to avoid a biased corpus, which might result in misleading generalizations.

For each of the experiments, we combined the available 92 metrics with ours to grasp the TBox aspect of ontologies, which is already achieved to some extent by Kang et al. (2014). Thus, we prepared two sets of metrics to train the models: 1) the originally available 92 metrics, and 2) 143 combined metrics (the previous set plus our newly proposed 51 metrics). Before training the models, we removed the metrics that had zero standard deviation.

After having analysed the results, we saw that the contribution of the metrics for the different datasets differed. This made us analyse the inner details of the used datasets to gain further insight on their distribution according to their size and ABox Intensivity. Finally, quality of the feature selection had also to be taken into consideration. Thus, we ran each of the experiments using the complete sets of metrics (*std* in the figures and tables), as well as using only the variables selected by the feature selection algorithm on each considered set (*SFA* in the figures and tables). This allowed us to see whether the feature selection algorithm was effective in our settings, and whether our newly proposed features were really contributing to the model while achieving the same or more prediction accuracy.

The R script code of all the experiments run (Random Forests based regression with 10-fold Cross Validation and 80%/20% validation, and feature selection algorithms), as well the results and the data analysis are available at[14].

Assessing the contribution of 51 ABox Metrics via 10-fold Cross Validation

Considering all the combinations datasets and sets of ontology metrics, we firstly trained our models and measured their prediction accuracy using 10-fold Cross Validation. The numeric results obtained are shown in Table 6, where the rows named *std* (standard) are the results produced without using feature selection; and the rows named SFA are the results obtained by only using the features that Boruta algorithm recommended.

[14] https://github.com/IsaGuclu/ReasoningABoxIntensiveOntologies

		92 Metrics	51 ABox + 92 Metrics
ELK DS1	std	7.62%	7.17%
	SFA	7.51%	7.42%
ELK DS2	std	5.32%	5.32%
	SFA	5.34%	5.30%
JFaCT DS1	std	46.94%	62.27%
	SFA	43.84%	53.48%
JFaCT DS2	std	6.38%	5.93%
	SFA	7.01%	6.10%
TrOWL DS1	std	131.76%	203.25%
	SFA	131.75%	140.09%
TrOWL DS2	std	37.25%	23.13%
	SFA	33.86%	23.02%

Table 6. 10-fold Cross Validated *MAPE* values

The graphs comparing R^2 and *MAPE* values are included in Figures 3 and 4, respectively (for the sake of clarity, values of Table 6 higher than 100 % have been truncated)

First, we will focus on *std* results, that is, those obtained using all metrics without feature selection:

- In the experiments with ELK using dataset-1, the changes in R^2 and *MAPE* between 92 metrics and combined metrics were very small, less than 1%. Using dataset-2, the changes in R^2 and *MAPE* between 92 metrics and combined metrics were even smaller than the ones with dataset-1, which was not a significant change.
- In the experiments with JFaCT using dataset-1, we observed an improvement in R^2 of ≈3% with the introduction of the new metrics, but it came along a ≈15% decrease in *MAPE*. Using dataset-2, R2 improved ≈1% and MAPE improved less than 1% with combined metrics.

Figure 3. Comparison of R^2 values in 10-fold Cross Validation procedure.

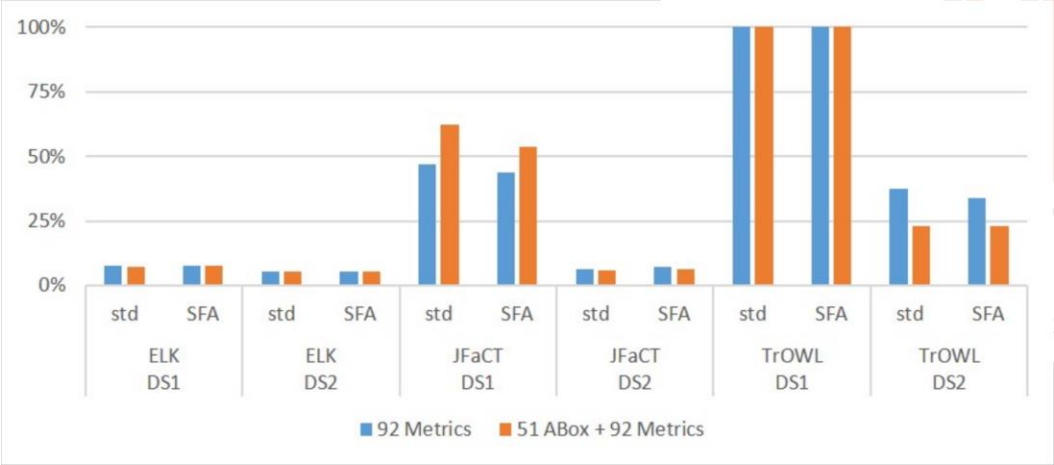

Figure 4. Comparison of *MAPE* values in 10-fold Cross Validation procedure.

- Finally, in the experiments with TrOWL using dataset-1, we observed a decrease in R^2 of ≈5% when using the combined metrics. Although the *MAPE* obtained with 92 metrics was better than the one obtained with 143 metrics, *MAPE* value for both were more than 100%, which is an undesirable for a prediction. Using dataset-2, we observed very similar R^2 with both metrics, but there was ≈14% improvement in *MAPE* (a relative improvement of ≈40%). This increase in prediction accuracy with the augmented dataset will be analysed later in the paper when we analyse the dataset distribution.

After getting this first set of results, we ran Boruta algorithm and obtained the metrics recommended in both cases (original and combined metrics). The details about the features selected by Boruta algorithm can be found in the Appendix, listed in Tables 8 and 9.

After having filtered the metrics sets selecting the features suggested by Boruta algorithm, we ran a new batch of the experiments which result are labelled as *SFA* in Table 6, and Figures 3 and 4:

- In the experiments with ELK using dataset-1, comparing with the *std* results, we observed an increase in R^2 of ≈5% and ≈12% for the original and combined metrics, respectively. Although there seemed not to be a significant difference at first when *std* metrics sets where considered, a difference emerged when feature selection algorithm was applied. Feature selection increased the accuracy of both metric sets, but in the case of combined metrics it showed ≈7% higher performance in terms of R^2. The difference in *MAPE* value was less than 1%. Using dataset-2, there was no notable change in $R^2$and *MAPE* between 92 metrics and combined metrics (the results for both sets are remarkable).
- In the experiments with JFaCT using dataset-1, comparing with the *std* results, we observed an increase in $R^2$of ≈ 4%, and an improvement in *MAPE* of ≈3% for the original metrics. For the combined metrics, there was no notable change in R^2, but an improvement in *MAPE* of ≈9%. Using dataset-2, comparing with the *std* results, we can see how the results got slightly worse (about a 1% in every dimension); however, they were still good accuracy values. When we compared the results the original and the combined metrics taking *std* and *SFA* methods into

consideration, we observe that combined metrics did not contribute to the prediction accuracy with dataset-1, but it showed a little contribution in dataset-2 with combined metrics. This made us search for the explanation of the change in the prediction accuracy on dataset-1 and dataset-2, which is analysed later along with the dataset distribution.

- Finally, in the experiments with TrOWL using dataset-1 (*SFA* row), we observed a decrease in R^2 of ≈2% when combined metrics were considered. Although the *MAPE* obtained with 92 metrics was better than the one obtained with 143 metrics, *MAPE* value for both were more than 100%, which is undesirable for a prediction model. Using dataset-2 (*SFA* row), we observed that the models obtained similar R^2values with both metrics, but there was an improvement in *MAPE* of ≈10% (a relative improvement of ≈30%). This increase in prediction accuracy with the augmented dataset will be analysed later along with the dataset distribution.

In general, the results we obtained by using feature selected showed a better (or similar in the worst case) prediction accuracy on the different datasets and reasoners considered in the experiments. From the perspective of datasets, we observed that combined metrics showed a contribution at different levels on dataset-2, but they did not so for dataset-1. As above mentioned, this will be analysed along with the dataset distribution.

Assessing the contribution of 51 ABox Metrics via 80%/20% separation

After we validated our models with 10-fold Cross Validation, we wanted to further test the prediction models in a more general scenario. To do so, we ran again the experiments randomly separating each of the datasets into a 80% training set, and a 20% testing set. We ran this experimental scenario 3 times and got the average of the values, which are shown in Table 7, and in Figures 5 and 6.

As we have done in the previous section, we used all available metrics in model generation and validation in the first batch of experiments (std in the table and figures). Then, we ran Boruta algorithm, selected the recommended metrics, and ran again the experiments in a second batch (SFA in the table and figures).

We can see how, using dataset-1, the combined metrics did not improve the precision of the results but for ELK reasoner, where they improved the R2 for the std setting. For dataset-2, if we focus on the contribution of the combined metrics, we see similar results as the analysis performed with 10-fold CV. The only remarkable situation is the increase of MAPE for TrOWL reasoner when using the std metric set; however, we have to bear in mind that the models in this schema were trained with less data (80% vs. 90% used in the 10-CV), and the number of repetitions is also lower (3 times vs. 10 times in the 10-CV), which indeed influenced the results.

Overall, these results in both 10-fold cross validation and 80%/20% separated validation support our previous observations.

		92 Metrics		51 ABox + 92 Metrics	
		R-Squared	MAPE	R-Squared	MAPE
ELK DS1	std	39.52%	9.10%	64.26%	6.58%
	SFA	53.15%	8.02%	55.77%	6.39%
ELK DS2	std	99.32%	5.23%	98.52%	5.28%
	SFA	98.94%	5.32%	98.55%	5.58%
JFaCT DS1	std	68.39%	44.73%	54.77%	23.35%
	SFA	82.68%	56.67%	78.14%	26.45%
JFaCT DS2	std	96.09%	7.12%	95.32%	6.63%
	SFA	95.99%	8.63%	96.13%	6.25%
TrOWL DS1	std	68.00%	86.31%	36.99%	167.59%
	SFA	94.10%	70.90%	81.15%	151.38%
TrOWL DS2	std	96.93%	21.00%	98.23%	35.64%
	SFA	98.18%	32.02%	98.09%	30.38%

Table 7. 80%/20% separated validation results

Figure 5. Comparison of R^2 values in the 80%/20% separated validation procedure.

Figure 6. Comparison of *MAPE* values in the 80%/20% separated validation procedure

ABox Intensity and Prediction Accuracy of Metrics

Although we were getting better results with the combined metrics than with the original ones on dataset-2, we could not get the same result on dataset-1. This made us question the difference between these two datasets. We analysed these datasets from the perspective of ABox/TBox ratio and ontology size. We grouped the ontologies into three levels of *ABox intensity* ($[0 - 5)$, $[5 - 10)$, $[10 - \infty)$); and into three different sizes: *small* (having up to 500 logical axioms), *medium* (having more than or equal to 500 logical axioms but less than 5000), and *big* ones (having more than or equal to 5000 logical axioms).

In Tables 4 and 5, we can see the distribution of the ontologies of dataset-1 and dataset-2, respectively. We can see how dataset-1 is clearly biased to ontologies with low ABox intensities, while dataset-2 (obtained by data augmentation) is not so biased to it. As a result of our observation and the results obtained in our experiments, we can claim that the metrics presented in this paper will improve the prediction accuracy when used on ABox intensive ontology sets.

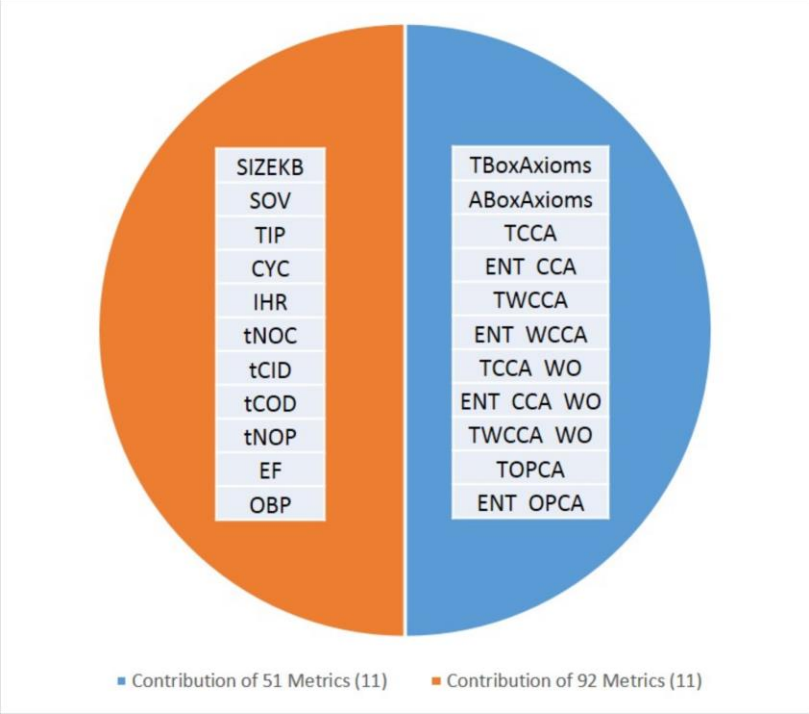

Figure 7. Metrics contributing to *ALL* (100%) of the models as selected by Boruta.

Assessing the Selection of Features by Boruta

Finally, we have to analyse the results we obtained by applying the Boruta feature selection algorithm to both the original metrics and the combined metrics sets[15] :

- When we searched for the metrics that are selected by Boruta algorithm in *all models* (, we saw that it selected 11 of the original 92 metrics, and 11 of the 51 newly proposed metrics were selected, as visualized in Figure 7. This implies that, on the one hand, there are original metrics

[15] The interested reader can find the details about the results of the selected metrics in the Appendix.

which are strongly relevant for the prediction task, but, on the other, that the newly included metrics are also contributing heavily to the model construction.

- When we search for the metrics that are selected as a related metric by Boruta *in at least 50% of the models*, we see that 34 of of 51 metrics and 27 of 92 metrics are selected.
- As a final note, the ABox intensivity[16] was selected as relevant for the three reasoners when facing an ABox intensive dataset (dataset-2).

This supports our hypothesis that the newly proposed metrics contribute to the prediction model as much as the available metrics.

DISCUSSION

Related Work

In our work, we have analyzed available metrics and investigated how to improve the capture of the complexity of ontologies by developing new metrics, which represent ABox axiom (and its interaction with TBox axioms) aspects of ontologies. According to our experiments comparing the original 92 metrics with our combined 143 metrics, we observe that adding the newly proposed metrics increases the accuracy of prediction in ABox intensive ontology sets.

From the results derived in our experiments, we can conclude that the previously available metrics (Kang et al. (2014)) could capture the complexity of ontologies to some extent. However, materialization needed new metrics that represented the interaction of ABox axioms with TBox axioms, and took their combined complexity into account. The amount of ABox axioms in ontology and their interactions can cause the consumption of more execution time than expected if their complexity is ignored. To fill such gap, we have proposed 51 new metrics to include the effect of ABox complexity in performance prediction of ontology reasoning, and we plan to improve these metrics further for more effectiveness.

Prediction Accuracy Change According to ABox Intensity

At the beginning of our research, we were wondering whether we could develop metrics that are generalizable regardless the characteristics of ontologies used in a particular scenario. After making our experiments, detailed in the previous section, we observed that it is *very difficult* to develop metrics that can show high prediction accuracy on all ontology sets.

In this work, we have observed and introduced the *ABox intensity* aspect of ontology sets. From our experiments, we have seen that this characteristic of the ontologies has an important influence on the prediction capabilities of the trained models. This observation along with the contribution of the newly proposed metrics developed according to this observation motivates us to look for deeper analysis on performance of ontology reasoning.

Feature selection

Apart from the increase in the accuracy of prediction, the identification of the metrics that actually

[16] Ratio metric in Table 8.

contribute to such improvement is also important. We have implemented the Boruta algorithm to select related features in model generation, and have seen observed that the models generated by the selected features either increase the accuracy of prediction or achieve similar results. Following this line of research, we plan to make further analysis with different feature selection algorithms in our future work.

We have analysed the features selected in the models generated for the considered reasoners (ELK, JFaCT and TrOWL) and datasets (dataset-1 and dataset-2). Every reasoner shows different reasoning behaviors according to the algorithms they are built on, and the optimization techniques they are using. Likewise, every ontology shows different reasoning requirements according to the domain of the knowledge modelled, expressivity of the selected language family, etc. This variability makes feature selection more and more important as new metrics will be introduced to grasp the complexity of ontology reasoning and make performance predictions with higher accuracy. Besides, decreasing the number metrics for prediction not only increases prediction accuracy, but also decrease resource consumption when calculating the metric values, which makes some computing tasks more applicable on resource bounded environments, such as mobile devices (Bobed et al. (2015), Krishnaswamy & Li (2014)).

CONCLUSION & FUTURE WORK

Performance prediction of ontology reasoning is a very interesting and challenging topic. In this work, we have introduced the concept of *ABox intensity*, showing that it has a strong influence on the predictability of the performance of ontology reasoning. Thus, we have focused our work on the performance prediction of ABox intensive OWL 2 EL ontologies, and proposed 51 new metrics which extend the previous work of Kang et al. (2014). The results obtained by adding these new metrics show an increase in the prediction accuracy of the trained model when dealing with ABox intensive ontology sets. Apart from the accuracy improvement, to see the contribution of the new metrics to the generation of prediction models, we also implemented a feature selection method, and saw that our new metrics contribute to the prediction model as much as the previously available metrics.

We believe that awareness of the *ABox intensity* in ontologies, and bringing a solution to propagate the complexity of the TBox to the ABox will increase the effectiveness and validity of prediction models on performance of ontology reasoning.

As for future work, firstly, we plan to work on better representations of the interactions between ABox axioms and TBox axioms by improving available metrics and extending them to the OWL 2 DL profile. Secondly, we will make experiments with more reasoners on different ontologies that will help understand the interaction of ABox axioms with TBox axioms in a broader sense. Finally, we will use different prediction mechanisms and feature selection algorithms to leverage the contribution of these metrics.

ACKNOWLEDGEMENTS

This work was partially supported by the EC Marie Curie IAPP K-Drive project (286348); projects TIN2013-46238- C4-4-R, TIN2016-78011-C4-3-R, and DGA-FSE; and the mobility research grant ”Programa Ibercaja-CAI de Estancias de Investigacio´ n” IT 22/16.

REFERENCES

Armas-Romero, A., Cuenca-Grau, B. & Horrocks, I. (2012), MORe: Modular combination of OWL reasoners for ontology classification, in 'Proceedings of the 11th International Semantic Web Conference (ISWC 2012), Part I', pp. 1–16.

Baader, F., Calvanese, D., McGuinness, D., Nardi, D. & Patel-Schneider, P. F. (2003), The Description Logic Handbook. Theory, Implementation and Applications, Cambridge University Press.

Bizer, C., Heath, T. & Berners-Lee, T. (2009), 'Linked Data - The story so far ', International Journal on Semantic Web and Information Systems 5(3), 1–22.

Bobed, C., Bobillo, F., Ilarri, S. & Mena, E. (2014), 'Answering continuous description logic queries: Managing static and volatile knowledge in ontologies', International Journal on Semantic Web and Information Systems 10(3), 1–44.

Bobed, C., Yus, R., Bobillo, F. & Mena, E. (2015), 'Semantic reasoning on mobile devices: Do androids dream of efficient reasoners?', Journal of Web Semantics 35(4), 167–183.

Burnham, K. P. & Anderson, D. R. (2002), Model Selection and Multimodel Inference: a Practical Information-theoretic Approach, Springer.

Burton-Jones, A., Storey, V. C., Sugumaran, V. & Ahluwalia, P. (2005), 'A semiotic metrics suite for assessing the quality of ontologies', Data & Knowledge Engineering 55(1), 84–102.

Cuenca-Grau, B., Horrocks, I., Motik, B., Parsia, B., Patel-Schneider, P. F. & Sattler, U. (2008), 'OWL 2: The next step for OWL', Journal of Web Semantics 6(4), 309–322.

Della Valle, E., Schlobach, S., Kro¨ tzsch, M., Bozzon, A., Ceri, S. & Horrocks, I. (2013), 'Order matters! harnessing a world of orderings for reasoning over massive data', Semantic Web 4(2), 219–231.

Dentler, K., Cornet, R., ten Teije, A. & de Keizer, N. (2011), 'Comparison of reasoners for large ontologies in the OWL 2 EL profile', Semantic Web 2(2), 71–87.

Fokoue, A., Meneguzzi, F., Sensoy, M. & Pan, J. Z. (2012), Querying Linked Ontological Data through Distributed Summarization, in 'Proceddings of the 26th AAAI Conference on Artificial Intelligence (AAAI 2012)', pp. 31–37.

Glimm, B., Horrocks, I., Motik, B., Stoilos, G. & Wang, Z. (2014), 'HermiT: An OWL 2 reasoner ', Journal of Automated Reasoning 53(3), 245–269.

Gonc¸alves, R. S., Bail, S., Jime´nez-Ruiz, E., Matentzoglu, N., Parsia, B., Glimm, B. & Kazakov, Y. (2013), OWL Reasoner Evaluation (ORE) workshop 2013 results: Short report, in 'Proceedings of the 2nd International Workshop on OWL Reasoner Evaluation (ORE 2013)', Vol. 1015, CEUR Workshop Proceedings, pp. 1–18.

Guclu, I., Bobed, C., Pan, J. Z., Kollingbaum, M. J. & Li, Y. (2016), How can reasoner performance of ABox intensive ontologies be predicted?, in 'Proceedings of the 6th Joint International Conference on Semantic Technology (JIST 2016)', pp. 3–14.

Guclu, I., Li, Y., Pan, J. Z. & Kollingbaum, M. J. (2016), Predicting energy consumption of ontology reasoning over mobile devices, in 'Proceedings of the 15th International Semantic Web Conference (ISWC 2016)', pp. 198–214.

Guo, Y., Pan, Z. & Heflin, J. (2005), 'LUBM: A benchmark for OWL knowledge base systems', Journal of Web Semantics 3(2–3), 158–182.

Hogan, A., Pan, J. Z., Polleres, A. & Ren, Y. (2011), Scalable OWL 2 Reasoning for Linked Data, in 'Tutorial Lectures of the 7th International Summer School 2011 (Reasoning Web 2011)'.

Hu, W., Chen, J., Zhang, H. & Qu, Y. (2011), How matchable are four thousand ontologies on the semantic web, in 'Extended Semantic Web Conference', Springer, pp. 290–304.

Hu, W., Qu, Y. & Cheng, G. (2008), 'Matching large ontologies: A divide-and-conquer approach', Data & Knowledge Engineering 67(1), 140–160.

Jolliffe, I. (2002), Principal component analysis, Springer

Kang, Y.-B., Li, Y.-F. & Krishnaswamy, S. (2012a), Predicting reasoning performance using ontology metrics, in 'Proceedings of the 11th International Semantic Web Conference (ISWC 2012)', pp. 198–214.

Kang, Y.-B., Li, Y.-F. & Krishnaswamy, S. (2012b), A rigorous characterization of classification performance - A tale of four reasoners, in 'Proceedings of the 1st International Workshop on OWL Reasoner Evaluation (ORE 2012)'.

Kang, Y.-B., Pan, J. Z., Krishnaswamy, S., Sawangphol, W. & Li, Y.-F. (2014), How long will it take? Accurate prediction of ontology reasoning performance, in 'Proceddings of the 28th AAAI Conference on Artificial Intelligence (AAAI 2014', pp. 80–86.

Kazakov, Y. (2008), SRIQ and SROIQ are harder than SH OIQ, in 'Proceedings of the 11th International Conference on Principles of Knowledge Representation and Reasoning (KR 2008)', pp. 274–284.

Kohavi, R. (1995), A study of cross-validation and bootstrap for accuracy estimation and model selection, in 'Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI 1995)', pp. 1137–1143.

Krishnaswamy, S. & Li, Y.-F. (2014), The mobile semantic web, in 'Proceedings of the 23rd International Conference on World Wide Web (WWW 2014)', ACM, pp. 197–198.

Kursa, M. & Rudnicki, W. (2010), 'Feature selection with the Boruta package', Journal of Statistical Software 36(1), 1–13. Matentzoglu, N., Bail, S. & Parsia, B. (2013), A corpus of OWL DL ontologies, in 'Proceedings of the 26th International Workshop on Description Logics (DL 2013)', pp. 829–841.

Maynard, D., Peters, W. & Li, Y. (2006), Metrics for evaluation of ontology-based information extraction, in 'Proceedings of the 4th International Workshop on Evaluation of Ontologies for the Web (EON 2006)'.

Ozkan, T. (2016), 'Reoffending among serious juvenile offenders: A developmental perspective', Journal of Criminal Justice 46, 18–31.

Pan, J. Z. (2004), Description Logics: reasoning support for the Semantic Web, PhD thesis, University of Manchester. Pan, J. Z., Ren, Y. & Zhao, Y. (2016), 'Tractable approximate deduction for OWL', Artificial Intelligence 235, 95–155. Pan, J. Z., Thomas, E., Ren, Y. & Taylor, S. (2012), 'Tractable fuzzy and crisp reasoning in ontology applications', IEEE Computational Intelligence Magazine 7, 45–53.

Ren, Y., Pan, J. Z. & Lee, K. (2012), Optimising parallel ABox reasoning of EL ontologies, in 'Proceedings of the 25th International Workshop on Description Logics (DL 2012)'.

Ren, Y., Pan, J. Z. & Zhao, Y. (2010), Soundness preserving approximation for TBox reasoning, in 'Proceedings of the 24th AAAI Conference on Artificial Intelligence (AAAI 2010)', pp. 351–356.

Sazonau, V., Sattler, U. & Brown, G. (2014), Predicting performance of OWL reasoners: Locally or globally?, in 'Proceedings of the 14th International Conference on Principles of Knowledge Representation and Reasoning (KR 2014)', pp. 661–664.

Sirin, E., Parsia, B., Cuenca-Grau, B., Kalyanpur, A. & Katz, Y. (2007), 'Pellet: A practical OWL-DL reasoner ', Journal of Web Semantics 5, 51–53.

Sleeman, D., Rissakis, M., Craw, S., Graner, N. & Sharma, S. (1995), 'Consultant-2: Pre-and post-processing of machine learning applications', International Journal of Human-Computer Studies 43(43—63), 907–928.

Thomas, E., Pan, J. Z. & Ren, Y. (2010), TrOWL: Tractable OWL 2 reasoning infrastructure, in 'Proceedings of the 7th Extended Semantic Web Conference (ESWC 2010), Part II', pp. 431–435.

Tsarkov, D. & Horrocks, I. (2006), FaCT++ description logic reasoner: System description, in 'Proceedings of the 3rd International Joint Conference on Automated Reasoning (IJCAR 2006)', pp. 292–297.

Urbani, J., Kotoulas, S., Maassen, J., van Harmelen, F. & Bal, H. E. (2010), OWL reasoning with WebPIE: calculating the closure of 100 billion triples, in 'Proceedings of the 7th Extended Semantic Web Conference (ESWC 2010), Part I', pp. 213–227.

Urbani, J., van Harmelen, F., Schlobach, S. & Bal, H. E. (2011), QueryPIE: Backward reasoning for OWL horst over very large knowledge bases, in 'Proceedings of the 10th International Conference on The Semantic Web (ISWC 2011), Part I', pp. 730–745.

van Harmelen, F. (2011), 10 Years of Semantic Web: does it work in theory? Keynote at the 10th International Semantic Web Conference (ISWC 2011). Retrieved January 13, 2017, from http:// www.cs.vu.nl/ ˜frankh/ spool/ ISWC2011Keynote/

W3C (2009), OWL 2 Web Ontology Language Profiles, W3C. Retrieved January 13, 2017, from http:// www.w3.org/ TR/ 2009/ REC-owl2-profiles-20091027/

Yao, H., Orme, A. M. & Etzkorn, L. (2005), 'Cohesion metrics for ontology design and application', Journal of Computer Science 1, 107–113.

Yus, R. & Pappachan, P. (2015), Are apps going semantic? A systematic review of semantic mobile applications, in 'Proceedings of the 1st International Workshop on Mobile Deployment of Semantic Technologies (MoDeST 2015)', pp. 2–13.

Zhang, H., Li, Y.-F. & Tan, H. B. K. (2010), 'Measuring design complexity of semantic web ontologies', Journal of Systems and Software 83, 803–814.

APPENDIX A. FEATURES SELECTED BY BORUTA

Tables 8 and 9 present the selected features in each for each of the datasets, reasoners, and metrics sets. We have excluded the metrics that have never been selected by the algorithm.

	ELK DS1	ELK DS2	JFaCT DS1	JFaCT DS2	TrOWL DS1	TrOWL DS2	#Sel
51 New Metrics	Comb.	Comb.	Comb.	Comb.	Comb.	Comb.	
TBoxAxioms	X	X	X	X	X	X	6
ABoxAxioms	X	X	X	X	X	X	6
Ratio		X	X	X		X	4
TCCA	X	X	X	X	X	X	6
AVG CCA		X	X	X			3
MAX CCA		X		X		X	3
STD CCA		X		X		X	3
ENT CCA	X	X	X	X	X	X	6
TWCCA	X	X	X	X	X	X	6
AVG WCCA		X		X		X	3
MAX WCCA		X		X	X	X	4
STD WCCA		X		X	X	X	4
ENT WCCA	X	X	X	X	X	X	6
TCCA WO	X	X	X	X	X	X	6
AVG CCA WO	X	X		X		X	4
MAX CCA WO		X	X	X		X	4
STD CCA WO		X	X	X		X	4
ENT CCA WO	X	X	X	X	X	X	6
TWCCA WO	X	X	X	X	X	X	6
AVG WCCA WO		X	X	X		X	4
MAX WCCA WO		X	X	X	X	X	5
STD WCCA WO		X		X	X	X	4
ENT WCCA WO		X	X	X	X	X	5
TOPCA	X	X	X	X	X	X	6
AVG OPCA		X		X	X	X	4
MAX OPCA		X	X	X	X	X	5
MIN OPCA	X						1
STD OPCA		X		X	X	X	4
ENT OPCA	X	X	X	X	X	X	6
TWOPCA		X	X	X	X	X	5
AVG WOPCA		X		X	X	X	4
MAX WOPCA		X	X	X	X	X	5
MIN WOPCA	X	X					2
STD WOPCA		X		X	X	X	4
ENT WOPCA		X	X	X		X	4
ENT DPCA			X				1
TWDPCA		X					1
AVG WDPCA		X					1
MAX WDPCA		X					1
ENT WDPCA		X	X	X			3

Table 8. Selection of Proposed Metrics by Boruta in Model Generation.

	ELK DS1	ELK DS2	JFaCT DS1	JFaCT DS2	TrOWL DS1	TrOWL DS2		ELK DS1	ELK DS2	JFaCT DS1	JFaCT DS2	TrOWL DS1	TrOWL DS2	
							#Sel.							#Sel.
92 Metrics	Mtr92	Mtr92	Mtr92	Mtr92	Mtr92	Mtr92		Comb.	Comb.	Comb.	Comb.	Comb.	Comb.	
SIZEKB	X	X	X	X	X	X	6	X	X	X	X	X	X	6
SOV	X	X	X	X	X	X	6	X	X	X	X	X	X	6
ENR		X	X	X	X	X	5		X		X		X	3
TIP	X	X	X	X	X	X	6	X	X	X	X	X	X	6
EOG		X	X	X	X	X	5		X		X	X	X	4
RCH		X	X	X	X	X	5		X		X			2
CYC	X	X	X	X	X	X	6	X	X	X	X	X	X	6
GCI	X	X	X	X	X	X	6		X	X	X	X		4
HGCI		X		X		X	3						X	1
ESUB		X	X	X	X	X	5		X					1
CSUB			X	X			2			X	X			2
SUPECHN	X	X	X	X	X	X	6	X	X	X	X		X	5
SUBECHN	X	X	X	X	X	X	6	X	X	X	X		X	5
SUBCCHN			X				1			X				1
DSUPECHN	X	X	X	X	X	X	6	X	X		X	X	X	5
DSUBECHN	X	X	X		X	X	5	X	X		X	X	X	5
DSUBCCHN			X				1			X				1
ELCLSPRT	X	X		X			3	X	X		X		X	4
ELAXPRT	X	X	X	X			4	X	X					2
HLC	X	X			X	X	4	X	X		X	X	X	5
RHLC		X		X		X	3		X		X		X	3
IHR	X	X	X	X	X	X	6	X	X	X	X	X	X	6
IND	X	X	X	X	X	X	6		X	X	X	X	X	5
aNOC		X	X	X	X	X	5		X	X	X			3
mNOC		X	X	X	X	X	5		X	X	X		X	4
tNOC	X	X	X	X	X	X	6	X	X	X	X	X	X	6
aCID		X	X	X	X	X	5		X		X		X	3
mCID		X	X	X	X	X	5		X	X	X	X	X	5
tCID	X	X	X	X	X	X	6	X	X	X	X	X	X	6
aCOD	X	X	X	X	X	X	6		X		X	X	X	4
mCOD		X	X	X	X	X	5		X				X	2
tCOD	X	X	X	X	X	X	6	X	X	X	X	X	X	6
aNOP		X	X	X	X	X	5		X	X	X		X	4
mNOP		X	X	X	X	X	5		X	X			X	3
tNOP	X	X	X	X	X	X	6	X	X	X	X	X	X	6
ENUM	X	X	X	X		X	5	X	X		X		X	4
ENUMP	X	X		X			3	X	X		X		X	4
CONJ	X	X	X	X	X	X	6		X	X	X		X	4
CONJP		X		X	X	X	4		X				X	2
EF	X	X	X	X	X	X	6	X	X	X	X	X	X	6
EFP		X			X	X	3		X				X	2
OBP	X	X	X	X	X	X	6	X	X	X	X	X	X	6
OBPP		X					1		X					1
DTP		X	X	X			3			X				1
DTPP		X					1		X					1
FUN		X				X	2							0
FUNP		X				X	2							0
TRN		X		X		X	3		X		X			2
TRNP		X		X		X	3		X		X			2
SUBP	X	X	X		X	X	5	X	X	X		X	X	5
DOMN		X					1	X						1
RANG		X		X			2		X		X			2
CHN		X		X	X	X	4		X		X	X	X	4

CHNP	X			X	X	3	X	X	X	3
ELPROP	X		X	X		3	X			1

Table 9. Selection of 92 Metrics by Boruta in Model Generation