$\mathcal{T} \cup \mathcal{X}(\alpha) \cup \mathcal{X}(\beta)$ is consistent for $(\alpha, \beta) \in \{(6, 9), (9, 9)\}$. All potential rules have a similar support and confidence of $1/2$, and all reach consequent (13) *which is consistent in any empty $\mathcal{Q}_5^9(9)$. $\mathcal{X}_5^9(9) \cup \mathcal{G} \not\models \perp$ since all entailments $\mathcal{G}$ of $\mathcal{X}_5^9$ are the same at time* 6 *and* 9. (13) *is the predicted knowledge.*

Streams	$\mathcal{O}_5^9$					$\mathcal{P}_5^9$		$\mathcal{Q}_5^9$	
Entailment $g \in \mathcal{G}$	(22)	(23)	(10)	(11)	(18)	(16)	(20)	(13)	(21)
Point of Time 9	✓		✓	✓			✓	?	?

Table 4: Support of Entailment $g \in \mathcal{G}$ in $\mathcal{O}_5^8$, $\mathcal{P}_5^8$, $\mathcal{Q}_5^8$.

Injected in $\mathcal{P}_0^n(n)$, predictions can be used by DL reasoners for deriving new entailments, as a side-effect of Algorithm 2.

Example 13. ***(Prediction Side Effects in Ontology Streams)*** *Applying GCI* (5) *and prediction* (13) *in Example 12 using DL reasoning reaches to bus delay information on road $\{r_3\}$.*

Although Examples 12 and 13 illustrate prediction where all entailments and rules are restricted to only $\{r_1\}$, Algorithm 2 (line 8) is designed for more complex and general cases.

5 Experimental Results

We report (i) scalability of our approach and (ii) accuracy of its results. In particular we study the impact of axioms, their consistency together with support, confidence, and auto-correlation thresholds on Algorithms 1 ($A1$ for rules selection), 2 ($A2$ for knowledge prediction). Our implementation is based on (i) an extension of InfoSphere Streams [Biem *et al.*, 2010] for processing ontology streams in real-time, coupled with (ii) CEL reasoner [Baader *et al.*, 2006] for standard DL reasoning, and (iii) an adaptation of Apriori [Agrawal and Srikant, 1994] supporting subsumption for determining association rules. For scalability reasons, rules are not injected in DL reasoning but only their consequents. The experiments have been conducted on a server of 4 Intel(R) Xeon(R) X5650, 2.67GHz cores, 6GB RAM.

• **Context:** Reputable live stream data (Table 5) related to road [a] weather conditions, [b] travel times, [c] incidents together with [d] bus GPS location, delay and congestion status in Dublin City has been considered. Besides an ontology of 55 concepts, 19 role descriptions (17 concepts subsume the 38 remaining ones with a depth of 3), we inject $14,316$ $\mathcal{EL}^{++}$ GCIs (through 6 RDF triples) to describe 4772 roads, their interconnections. The objective is to predict which buses (among 300 buses) in [d] will be delayed in the next hour, using cross-stream rules selected from $A1$ and $A2$.

DataSet	Size (Mb) per day	Frequency of Update (seconds)	#Axioms per Update	#RDF Triples per Update
[a] Weather	3	300	53	318
[b] Travel	43	60	270	810
[c] Incident	0.1	600	81	324
[d] Bus	120	40	3,000	12,000

Table 5: Stream Datasets Details (average figures).

• **Scalability Result:** Figure 6 reports scalability of $A1$, $A2$ ((m_a, m_s, m_c) being $(0, 1/2, 1/2)$) and compares its computation time with a state-of-the-art approach [Wang *et al.*, 2003] in stream prediction, noted [W03]. They solve a classification problem over sensor raw data using statistics-based data mining techniques. The evaluation is achieved on (i) various sizes of stream windows $|w|$ (for learning/training) i.e., $\{1, 12, 48\}$ hours, and (ii) different number of streams $|s|$ i.e., $\{1, 3, 4\}$ for respectively [d], [b,c,d], [a,b,c,d] in Table 5.

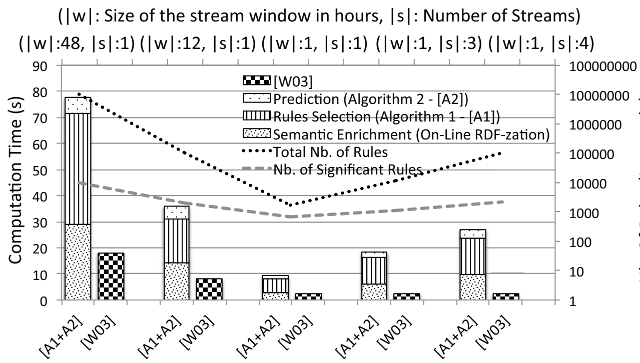

Figure 6: Scalability of Prediction Computation.

Unsurprisingly, [W03] scales much better than our approach in all contexts. Our approach requires some non-negligible computation time for the semantic enrichment of streams. In addition, as the number of potential rules is exponential with the number of entailments in streams (secondary vertical axis), the identification of significant rules is time consuming specially when the window size is growing. Once all rules are identified, the pure prediction part performs from $1.1s$ to $6.2s$. As [W03] is mainly designed for one stream, the computation time remains unchanged if multiple streams are considered.

• **Accuracy Result:** Figure 7 reports the prediction accuracy of both approaches where Table 6 is used only to configure the parameters values (m_a, m_c, m_s) of our approach $A2$. Prediction is computed within a window of 48 hours with all streams (Table 5). Accuracy is measured by comparing predictions (delayed buses) with real-time situations in Dublin City, where results can be easily extracted and compared from the raw and semantic data in respectively [W03] and our approach. Negative auto-correlation (c_1-c_4) strongly alters the accuracy while support and confidence have a positive effect. The confidence has a stronger impact on (i) the reduction of significant rules and (ii) accuracy. If positive auto-correlation, the accuracy of $A2$ (with a minimum confidence and support of .4) results outperforms results of [W03].

	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8
m_a	< 0				> 0			
m_c	.4	.4	.8	.8	.4	.4	.8	.8
m_s	.4	.8	.4	.8	.4	.8	.4	.8

Table 6: (m_a, m_c, m_s) Configuration.

• **Lessons Learnt:** Although inferring cross-streams rules has a positive impact on consistent prediction and its accuracy, it alters its computation time specially if streams derived numerous entailments. It is even worst with more expressive DLs because of the auto-correlation evaluation (line 7 in Algorithm 1) and consistency check (line 10 in Algo-