ALLDATA 2020

The Sixth International Conference on Big Data, Small Data, Linked Data and Open Data

ISBN: 978-1-61208-775-7

February 23 - 27, 2020

Lisbon, Portugal

ALLDATA 2020 Editors

Jedrzej Rybicki, Forschungszentrum Juelich GmbH, Germany

ALLDATA 2020

Forward

The Sixth International Conference on Big Data, Small Data, Linked Data and Open Data (ALLDATA 2020), held between February 23-27, 2020 in Lisbon, Portugal, continued a series of events bridging the concepts and the communities devoted to each of data categories for a better understanding of data semantics and their use, by taking advantage from the development of Semantic Web, Deep Web, Internet, non-SQL and SQL structures, progresses in data processing, and the new tendency for acceptance of open environments.

The volume and the complexity of available information overwhelm human and computing resources. Several approaches, technologies and tools are dealing with different types of data when searching, mining, learning and managing existing and increasingly growing information. From understanding Small data, the academia and industry recently embraced Big data, Linked data, and Open data. Each of these concepts carries specific foundations, algorithms and techniques, and is suitable and successful for different kinds of application. While approaching each concept from a silo point of view allows a better understanding (and potential optimization), no application or service can be developed without considering all data types mentioned above.

We welcomed academic, research and industry contributions. The conference had the following tracks:

- Big Data
- Open Data
- Linked Data
- Challenges in processing Big Data and applications

We take here the opportunity to warmly thank all the members of the ALLDATA 2020 technical program committee, as well as all the reviewers. The creation of such a high quality conference program would not have been possible without their involvement. We also kindly thank all the authors who dedicated much of their time and effort to contribute to ALLDATA 2020. We truly believe that, thanks to all these efforts, the final conference program consisted of top quality contributions.

We also thank the members of the ALLDATA 2020 organizing committee for their help in handling the logistics and for their work that made this professional meeting a success.

We hope that ALLDATA 2020 was a successful international forum for the exchange of ideas and results between academia and industry and to promote further progress in the field of all data. We also hope that Lisbon, Portugal provided a pleasant environment during the conference and everyone saved some time to enjoy the historic charm of the city.

ALLDATA 2020 Chairs

ALLDATA Steering Committee

Venkat N. Gudivada, East Carolina University, USA
Sandjai Bhulai, Vrije Universiteit Amsterdam, the Netherlands
Jerzy Grzymala-Busse, University of Kansas, USA
Fabrice Mourlin, Université Paris-Est Créteil Val de Marne, France

ALLDATA Industry/Research Advisory Committee

Stephane Puechmorel, ENAC, France

Hanmin Jung [정한민], Korea Institute of Science and Technology Information, South Korea

ALLDATA 2020

Committee

ALLDATA Steering Committee

Venkat N. Gudivada, East Carolina University, USA
Sandjai Bhulai, Vrije Universiteit Amsterdam, the Netherlands
Jerzy Grzymala-Busse, University of Kansas, USA
Fabrice Mourlin, Université Paris-Est Créteil Val de Marne, France

ALLDATA Industry/Research Advisory Committee

Stephane Puechmorel, ENAC, France
Hanmin Jung [정한민], Korea Institute of Science and Technology Information, South Korea

ALLDATA 2020 Technical Program Committee

Hugo Alatrista-Salas, Universidad del Pacífico, Peru
Houda Bakir, Datavora, Tunisia
Gábor Bella, University of Trento, Italy
Sandjai Bhulai, Vrije Universiteit Amsterdam, Netherlands
Jean-Yves Blaise, CNRS (French National Centre for Scientific Research) | UMR CNRS/MC 3495 MAP, France
Didem Gurdur Broo, KTH - Royal Institute of Technology, Sweden
Ozgu Can, Ege University, Turkey
Rachid Chelouah, Ecole Internationale des Sciences du Traitement de l'Information (EISTI), Cergy, France
Esma Nur Cinicioglu, Istanbul University - School of Business, Turkey
Cinzia Daraio, Sapienza University of Rome, Italy
Maaike de Boer, TNO, Netherlands
Bidur Devkota, Asian Institute of Technology (AIT), Thailand
Christian Dirschl, Wolters Kluwer Deutschland GmbH, Germany
Ricardo Eito Brun, Universidad Carlos III de Madrid, Spain
Mounim A. El Yacoubi, Telecom SudParis / Institut Mines Telecom / Institut Polytechnique de Paris, France
Aniekan Essien, Swansea University, UK
Hadi Fanaee-T, University of Oslo, Norway
Denise Beatriz Ferrari, Instituto Tecnológico de Aeronáutica, São José dos Campos - SP, Brazil
Panorea Gaitanou, University of Alcala, Spain
Fausto Pedro Garcia Marquez, University of Castilla-La Mancha, Spain
Raji Ghawi, Technical University of Munich, Germany
Jerzy Grzymala-Busse, University of Kansas, USA
Venkat N. Gudivada, East Carolina University, USA
Samrat Gupta, Indian Institute of Management Ahmedabad, India
Qiwei Han, Nova School of Business & Economics, Portugal

Copyright Information

For your reference, this is the text governing the copyright release for material published by IARIA.

The copyright release is a transfer of publication rights, which allows IARIA and its partners to drive the dissemination of the published material. This allows IARIA to give articles increased visibility via distribution, inclusion in libraries, and arrangements for submission to indexes.

I, the undersigned, declare that the article is original, and that I represent the authors of this article in the copyright release matters. If this work has been done as work-for-hire, I have obtained all necessary clearances to execute a copyright release. I hereby irrevocably transfer exclusive copyright for this material to IARIA. I give IARIA permission or reproduce the work in any media format such as, but not limited to, print, digital, or electronic. I give IARIA permission to distribute the materials without restriction to any institutions or individuals. I give IARIA permission to submit the work for inclusion in article repositories as IARIA sees fit.

I, the undersigned, declare that to the best of my knowledge, the article is does not contain libelous or otherwise unlawful contents or invading the right of privacy or infringing on a proprietary right.

Following the copyright release, any circulated version of the article must bear the copyright notice and any header and footer information that IARIA applies to the published article.

IARIA grants royalty-free permission to the authors to disseminate the work, under the above provisions, for any academic, commercial, or industrial use. IARIA grants royalty-free permission to any individuals or institutions to make the article available electronically, online, or in print.

IARIA acknowledges that rights to any algorithm, process, procedure, apparatus, or articles of manufacture remain with the authors and their employers.

I, the undersigned, understand that IARIA will not be liable, in contract, tort (including, without limitation, negligence), pre-contract or other representations (other than fraudulent misrepresentations) or otherwise in connection with the publication of my work.

Exception to the above is made for work-for-hire performed while employed by the government. In that case, copyright to the material remains with the said government. The rightful owners (authors and government entity) grant unlimited and unrestricted permission to IARIA, IARIA's contractors, and IARIA's partners to further distribute the work.

Table of Contents

Development and Implementation of an Ontology to Support the Product Development of Smart Textiles Using Open Innovation Platforms

Inga Gehrke*, Magnus Knuth†‡, Sabine Kolvenbach§, Urs Riedlinger§, Thomas Gries* and Sebastian Tramp†
*Institut fuer Textiltechnik at RWTH Aachen University, Germany, Email: inga.gehrke@ita.rwth-aachen.de
†eccenca GmbH, Leipzig, Germany, Email: {magnus.knuth,sebastian.tramp}@eccenca.com
‡Karlsruhe Institute of Technology, Germany, Email: magnus.knuth@kit.edu
§Fraunhofer FIT, Sankt Augustin, Germany, Email: {sabine.kolvenbach,urs.riedlinger}@fit.fraunhofer.de

***Abstract*—To create innovative connected products and services, more and more interdisciplinary development efforts across industries covering hardware, software and business model design are required. One example for this is smart textiles, where the complexity of the value chain has so far hindered the successful market launch of new products. In this paper, semantic web ontologies are used to support faster development and market entry by a structured interaction of all players along the value chain. A specific *Smart Textiles Ontology* is defined, validated and evaluated with the help of the structured incorporation of expert knowledge. The ontology acts as a foundation for an open co-innovation platform called GeniusTex, and has successfully enabled first product development projects.**

Keywords–Ontology Development; Smart Textiles.

I. INTRODUCTION

Smart Textiles are textiles with an extended range of functions that enable the interaction of the textile with the environment and the human user. Applications range from shirts for monitoring vital signs, activity or stress in health and sports to displays integrated in furniture and protective and heating equipment in industrial settings [1]. A standard of the European Committee for Standardization (CEN) defines *Smart Textiles* more specifically as intelligent systems consisting of textile and non-textile components that actively interact with their environment, a user or an object. Data is recorded and processed via sensors and a defined reaction is generated via actuators or an information display on a supplementary device [2].

The market outlook for smart textiles is very promising, with a market size of approximately €5 billion for 2022, growing from €1.3 billion in 2017 [3]. However, this potential has not yet been realized and few products have passed from prototype stage to mass market. Challenges arise mainly due to the complex value chain of smart textiles that includes textile and electronics suppliers, software and application developers, product designers and manufacturers (system integrators), as described also by [1], [4]–[6]. Such challenges are:

- *Technical challenges* that include the complex design of interfaces between textile and electronics components, the miniaturisation of all components for the seamless integration in textiles, as well as their usability and durability. To solve these challenges, expertise from different knowledge domains, such as textiles and electronics production, needs to be combined during the Product Development Process (PDP).
- *Organizational challenges* that emerge from diverse value chain which has limited experience in collaboration across the different knowledge domains. Due to a lack of standards in the smart textiles domain, the producers cannot refer to norms to ensure interoperability between their domains, but need to collaborate intensely with all parties along the value chain during the PDP.

This paper describes how the GeniusTex project addresses these complex challenges by utilising a *Smart Textiles Ontology*, which can be retrieved from [7], together with a co-innovation platform.

- Related work on ontologies for collaborative product development and (smart) textiles (Section II)
- Defining an Ontology as common language to describe the modules and diverse knowledge areas of smart textile domain (Section III)
- Validation (Section IV) and implementation on a co-innovation platform called GeniusTex (Section V)

II. RELATED WORK

Product and process development needs to handle a demand for shortened and more efficient development cycles despite higher product complexity [8]. Modularization is an established method to handle the complexity of product development and production processes by dividing the product into modules with clearly defined interfaces and relationships. This structured representation of the product enables both parallel development of modules, as well as reusing previously developed modules in new products [9]. Plus, a modular product concept simplifies collaborative development projects that involve different players along the value chain. While this has been the standard approach for collaborating both in development and production for industries like automotive, where Original Equipment Manufacturers (OEMs) and suppliers are highly integrated, it has not been adopted yet in the textiles domain [10]. Small and Medium Enterprises (SMEs) lack the integrated supply chain, especially to cover both textile, electronics and software components required for smart textiles.

Knowledge management and information sharing across all parties involved has been identified as a crucial prerequisite for successful development processes [11] [12]. Ontologies are one concept to support this by acting as a ”common language” between modules and parties involved in PDP. An ontology has been defined by Gruber [13] as a formal, explicit specification of a shared conceptualisation. Ontologies support knowledge management by structuring knowledge domains

 ISBN: 978-1-61208-775-7

into classes with properties and relations, thus allowing for searches and processes to be defined along the structures [10]. Since ontologies can be expanded, adapted and merged, they can support knowledge domains subject to changes in technological advances [14].

A. Ontologies enabling collaboration along the value chain

The use of ontologies to support the PDP has been described in literature for various knowledge domains, for instance to promote knowledge integration and sharing [12]. In addition to the mere provision of knowledge, ontologies can also enable decision-support in the PDP, e. g. by automatically finding previously developed modules or documents with high similarity to the current product specification [15], or suggesting new combinations of prior modules into new products [16]. Moreover, ontologies have been applied with a focus on collaboration and co-innovation. The collaborative product innovation network by Song et al. applies ontology based knowledge integration to reduce the development time of a water heater [17]. To integrate knowledge across different players along an aviation development process, Li et al. merge and map local ontologies into a global ontology for knowledge sharing [18].

B. Ontologies for Textiles

For textiles, there are few examples of structured knowledge management using ontologies. A knowledge exchange infrastructure has been proposed to promote collaboration [19] and to support interoperability in the textile industry [10]. The *Textile chemical ontology* focusses on hazardous and forbidden chemicals for textiles, e. g. by formalizing standards like the OEKO-TEX label [20]. While the *VetiVoc* ontology addresses only the fashion domain rather than technical or smart textiles, it gives an example how textiles can be modularized [21].

C. Ontologies for Smart Textiles and Services

In the domain of smart textiles and services, ontologies have been applied mainly for integrating different data sources. For example, sensor data collected by a wearable are combined with historical health data and weather data for health care services [22] or used to process wearable sensor data into specific human gestures [23]. The ontology-driven open reference architecture and platform around services for elderly of the project *OASIS* combines multiple measurements from smart textiles for unobtrusive monitoring applications [24].

However, the idea of exploiting modularity to handle the complexity of smart textile development processes has been introduced for example by Schwarz et al. [25]. They describe 6 functions of an intelligent textile system whose interdependence needs to be accounted for during the PDP, and which act as foundation for the functional components of a smart textile introduced in Section III. Similarly, reusing certain modules for multiple products is promising to reduce time and costs of smart textiles development. The *Simple Skin project* proposes textile building blocks where a conductive fabric acts as foundation for various products and services [26]. The *EASY-IMP project* described the use of meta-products that can be enhanced with different modules and services [27].

This research aims at exploiting the potential of ontologies to support the PDP of smart textiles. To address their specific challenges and the gaps in previous research, the Smart Textiles Ontology focuses on two objectives:

1) structured and accessible knowledge base across the smart textile domain
2) enabler of collaborative development and co-innovation processes

III. Defining the Smart Textiles Ontology

The domain of smart textiles covers multiple areas, such as textile materials, electronic components, as well as manifold steps of the production process. Hence, an ontology for smart textiles should cover these concepts and allow to express interdependencies among them in order to direct development projects. The scope of the ontology is determined by a set of natural language competency questions that are expected to be answered by the ontology in the end.

The smart textiles domain is described in modular structure, looking at classes of processes, materials, and component modules. Component modules include the functional parts that make a textile smart (sensors, actuators), data processing, data transmission and interconnection modules, as well as textile carrier or substrate. The production processes cover joining, separating, forming, handling and quality control. The focus lies on the joining class, as it includes the critical processes of integrating functional components to a textile carrier and the contacting between different components. For all these modules, properties and relations that describe the interdependencies between materials, components and processes are defined. This allows for the codification of domain knowledge specific to smart textiles: e. g., the drapability and elasticity of components is a critical property. While it is common to account for this in textile processing, for developers with an electronics background it is crucial to know this to select both electronic components and technologies for handling and integrating them accordingly.

A common challenge for any ontology development project is the effective involvement of domain experts. Specialists in related fields possess knowledge and skills that are indispensable for the construction of a domain ontology. Whereas they neither have to be experts in ontology development nor in the rather technical tools, the contributing knowledge engineers are often no experts in the domain. To overcome this problem in the GeniusTex project, the domain experts are provided with a minimal knowledge acquisition template, which is familiar to them. In this particular case, we use an Excel master document for collecting the ontology elements. There, the domain experts simply have to fill in labels, definitions, and straightforward references (e. g., superclass, domain, and range) for *Classes*, *Relations*, *Attributes*, and *Individuals*. This still demands a fundamental understanding of what an ontology is, but keeps away the technicalities of developing an OWL ontology.

The actual ontology is automatically built by the *eccenca Ontology Pipeline* based on the master document and the configurations made by a supporting knowledge engineer. The domain experts get immediate feedback in form of a validation report and an ontology visualisation (c. f. Section IV). The eccenca ontology pipeline is integrated in a continuous integration pipeline, which regenerates all relevant artifacts on updating the master document. With each commit the pipeline (1) generates an OWL file, (2) validates the ontology file resulting in a validation report, (3) creates a corresponding visualisation and (4) an ontology documentation. The WIDOCO tool [28] is used in this setting. These artifacts

 ISBN: 978-1-61208-775-7

are immediately accessible to the editors (domain experts and knowledge engineer), i. e., with each commit the contributors get information whether the ontology was build successfully and could be validated, as well as a visual representation.

According to the elements defined by the domain experts, a knowledge engineer solves particular ontological problems and enforces best practices, e. g. mapping of external ontologies, realising consistent structures and naming conventions. He also defines example instance data which is used to validate the ontology.

The GeniusTex Smart Textiles Ontology reuses concepts from multiple external ontologies, such as

- the *Semantic Sensor Network Ontology* (SSN) for the description of sensor features [29],
- the *Ontology of Units of Measure 2.0* for representing units of measurement needed to describe sensor features [30], whereas a number of very specific units had to be replenished for the textile domain, as well as
- *schema.org* for general product aspects of individual components, e. g. dimensions and price.

IV. Validation and Evaluation of the Smart Textiles Ontology

To ensure an adequate level of quality throughout the ontology development process, routines have been set up to check for multiple validation constraints after each development step. Finally, the Smart Textiles Ontology has been evaluated with actual data.

A. Validation

During the ontology development, it is crucial to validate the ontology after each development step in an automated way in order to find problems and quality issues fast. There are several criteria to validate an ontology. Some of these criteria can be checked fully automatically, such as syntactical correctness, logical consistency, and the application of some best practices. Other criteria demand judgement from experts who have a certain domain knowledge. For example whether the ontology is coherent to a common understanding of the domain.

We are following this approach with a continuous integration ontology development pipeline, which generates reports on each revision of the ontology. The aim is to evaluate and report as many things a possible in an automated way. These reports are played back to the domain experts who are obliged to keep the ontology in a validated state with every change. The *eccenca Ontology Pipeline* is based on a version control system and a continuous-integration (CI) server. The build process and integration tests are run automatically on the CI server when a new commit has been made:

1) The build process compiles the ontology from a knowledge acquisition template and several configurable source files.
2) An RDF parser (in this case, Apache Jena RIOT) ensures syntactical correctness.
3) An OWL reasoner (in this case, Pellet [31]) ensures logical consistency.

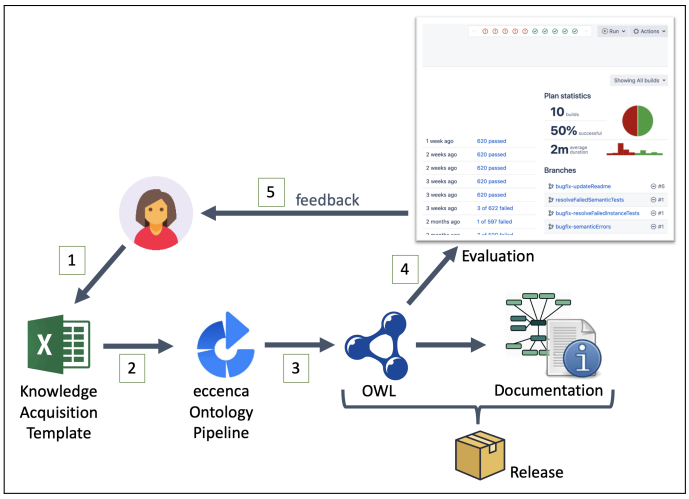

Figure 1: Continuous Integration Workflow.

4) RDFUnit [32] ensures a number of general best practice requirements, which are defined as tests in form of SPARQL queries or SHACL shapes.
5) OWL2DOT generates a visual representation of the ontology elements (see Figure 2).
6) WIDOCO generates an ontology documentation HTML document explaining the ontology elements.

The overall workflow is depicted in Figure 1. The results of all tests are collected as build artefacts on the CI server and respective authors (committers) are automatically informed if one of the tests fails. Furthermore, the generated documentation allows domain experts to inspect the defined elements in order to spot problems and share the current state of the ontology with colleagues. RDFUnit generates tests from the involved ontologies and executes these tests on the ontology definition and the optionally given instance data. The best practice requirements that are currently configured are mainly targeting documentation aspects, whereas it is desirable to extend these in the future:

- All defined resources should have an rdf:type
- All defined resources should have at least one label and a comment
- All defined resources should have different comments in different languages
- All defined resources should have a label different from its comment
- All labels and comments must not contain *todo*, *foo*, *bar*, *lorem* or *ipsum*
- Two defined resources should not have the same comment
- Two defined resources should not have the same label
- All resources defined by an ontology should be prefixed with the ontology URI and link this ontology with rdfs:isDefinedBy
- All ontologies should state their preferred namespace prefix
- All ontologies should state their preferred namespace matching their URI

 ISBN: 978-1-61208-775-7

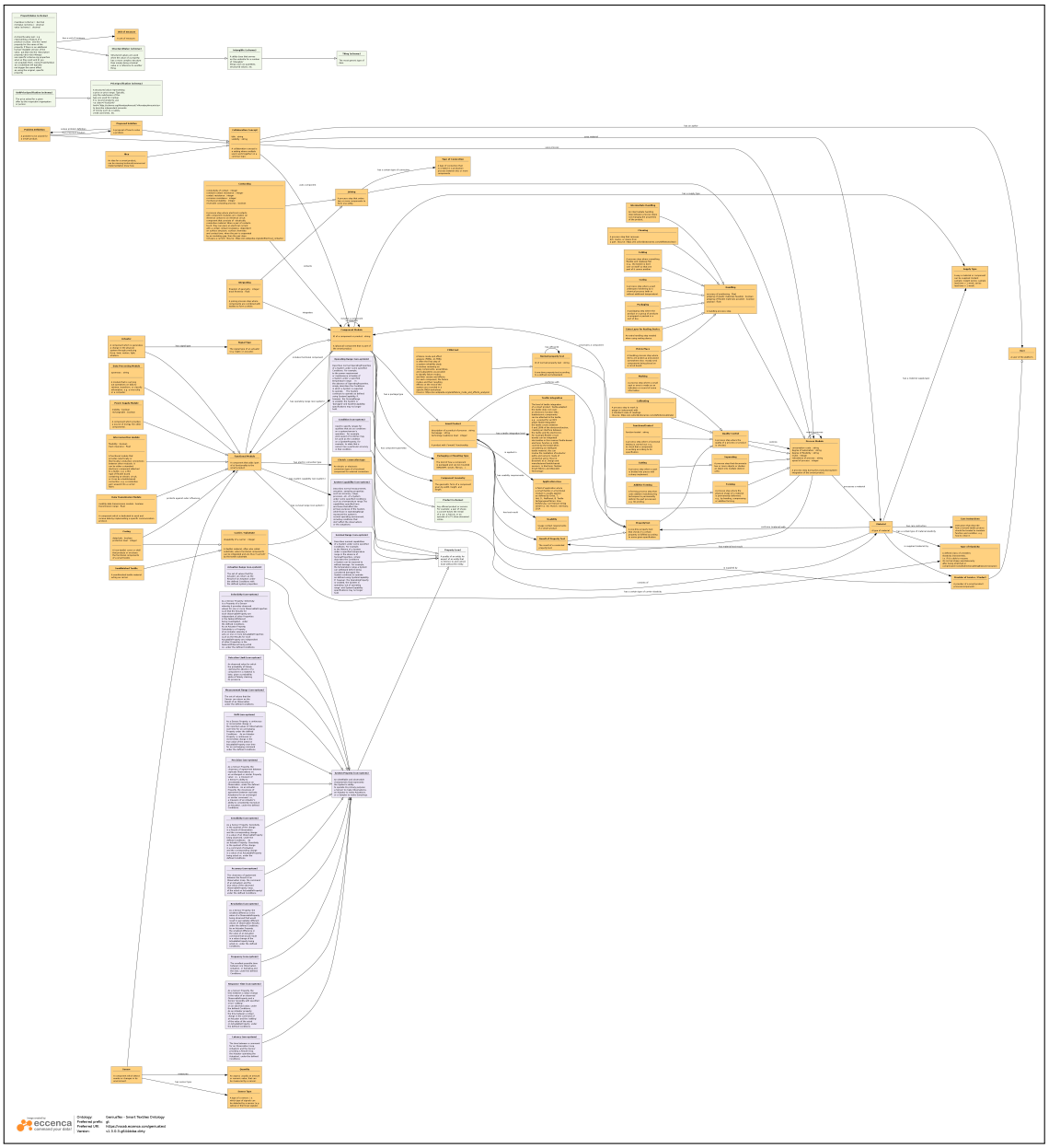

Figure 2: A visual representation of the Smart Textiles Ontology.

- All ontologies should have a version info

By adhering to these concepts, it can be made sure that the ontology remains valid and operational during its development cycle. Secondly, it leads to a more consistent development in general and an overall quality improvement.

B. Evaluation

In order to evaluate the Smart Textiles Ontology, a graph had to be populated with respective A-Box data. This has been done based on actual data on smart textile components along two example products (smart orthosis, smart wristband) as it occurred in the project. The data has been transformed to RDF using eccenca's Corporate Memory data integration capabilities. While doing so, the resulting triples have been continuously validated against the ontology.

For evaluating the ontology, 27 (out of originally 35) competency questions have been translated to SPARQL queries and finally executed on our knowledge graph. The questions have been collected at the beginning of the project, so that eight questions were not covered by the ontology in the end, mainly due to data gaps (e. g., *CQ 3.4 Which companies can create a prototype with SMD (surface mounted device)?*) or an no further pursued meta-level (e. g., *CQ 3.7 What is generally required for approval of a smart textile?*).

An example query for *CQ 2.11 Which actuators, which emit light or acoustic signals, are smaller than 20x20x3mm?* can be seen in Listing 1, it filters actuators by their signal type and physical dimensions. Since for all dimensions the millimetre unit is used, it has been neglected here (c. f. Section VI-A).

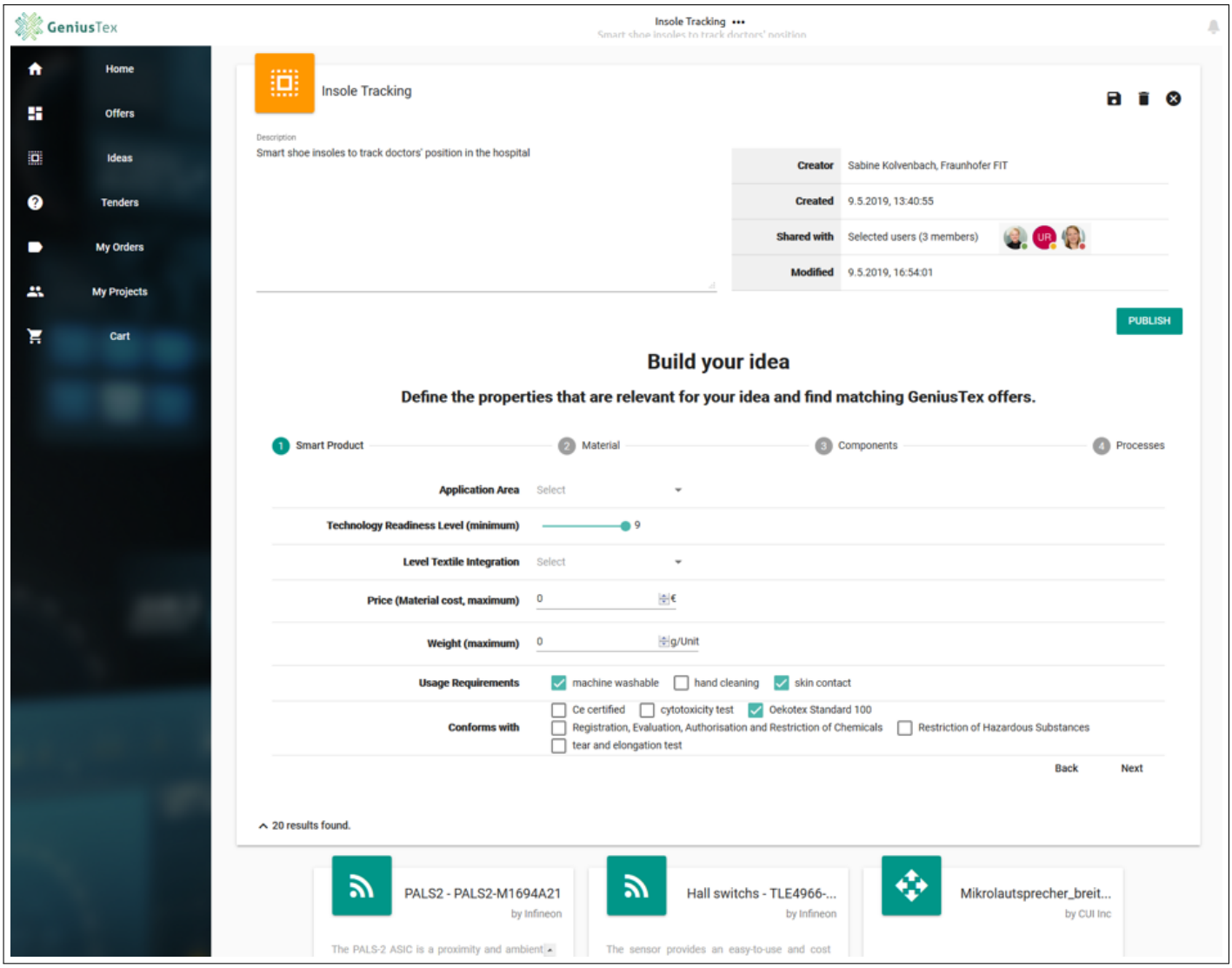

Figure 3: The collaborative GeniusTex Idea Configurator.

Listing 1: Query

```
<SELECT * WHERE {
  ?act a gt:Actuator ;
    gt:hasSignalType ?signal ;
    schema:depth [schema:value ?d] ;
    schema:width [schema:value ?w] ;
    schema:height [schema:value ?h] .
  FILTER (?signal = gt:opticsignal ||
          ?signal = gt:acusticsignal)
  FILTER (?d <= 20 && ?w <= 20 && ?h <= 3)
}
```

V. Implementation on GeniusTex - current infrastructure

Beyond the modularization of smart textile products, services and processes with the Smart Textiles Ontology, GeniusTex addresses the challenges of the smart textiles PDP with an open innovation platform. The platform enhances the PDP along the smart textile value chain by enabling all involved stakeholders such as manufacturers, suppliers, service-providers, and end-users to connect, collaborate, ideate, and develop innovate business models. Main features of the GeniusTex platform are the import of ontology-based product data and its semantic representation, intelligent search to find and order relevant and compatible smart textile products, ideas browsing and collaborative idea configuration as well as information sharing in protected workspaces to generate new knowledge with experts and end-users.

Figure 3 shows the shared idea “Insole Tracking”. The three idea members can collaboratively build the idea by defining the properties that are relevant for the perspective smart textile product. In this example, authors are interested in smart products with the *TRL 9* (highest level of technology readiness), which have to be machine washable, are designed for skin-contact, and conforms to *Oekotex Standard 100*. GeniusTex finds twenty results that match the configuration. Idea members can browse the results, find detailed product information and order suitable products. Moreover, users can

publish the idea to the GeniusTex community with intend to find and discuss with other experts. GeniusTex platform users get notified of the published idea and can ask for joining the PDP.

VI. Conclusion and Future Work

The application of an ontology has been proven as an eligible common language in the multi-disciplinary field of smart textile development. The steep learning curve that comes along the "ontological overhead" faced by many domain experts confronted with an ontology development project, could be obviated by providing them familiar tools and immediate feedback mechanisms. Nevertheless, some decisions have to be made by knowledge engineers who are aware of the technological pitfalls.

A. Learnings: success factors for applying the Smart Textiles Ontology

When creating the Smart Textiles Ontology, a number of existing concepts from external ontologies have been reused, which is in particular helpful for compatibility reasons. Nevertheless, we experienced some issues which are generally unresolved: There are plenty of ways, but no standard pattern to represent measurements and physical quantities in RDF, i. e. values with well-defined units. We decided to go for schema.org's `PropertyValue` pattern, mainly because it supports values and value ranges along with units of measurements, for which the Ontology of units of Measure (OM) is used. While this approach allows to explicitly express values, it became apparent that it is on the other hand intricate to write queries for that model. Furthermore, it would actually be necessary to convert units of measurement during the query process to compare between measurements using different units, e. g. as suggested by Lefrancois using Custom Datatypes [33]. To really benefit from this variety of representation possibilities it would be beneficial to have the ability to transform between standard representations. Likewise, even though multiple ontologies for units of measurement exist, none is fully covering units from all domains [34]. Consequently, a number of units for the textile domain were added to the OM ontology.

B. Outlook: Internationalising and broadening the functionalities of GeniusTex

The Smart Textiles Ontology and its implementation on the GeniusTex platform help to streamline smart textile product and process development. So far, it has enabled three co-innovative development projects across different application areas: a smart orthosis, a smart wristband, and a connected pillow. To better support future projects, further functionalities have been identified, such as the link to broader product databases, modelling of complex inter-dependencies and rules within the ontology and improving search and recommendation functions based on this. Moreover, the internationalisation of the ontology and the platform is a next step. To ensure both platform architecture and its structured language are globally accessible, workshops with partners from different linguistic backgrounds, e. g. from South Korea, are conducted.

Finally, the platform-based development approach enabled by an ontology is applicable beyond the smart textiles domain. It is relevant whenever interdisciplinary domain knowledge from suppliers, manufacturers and service providers from different industries needs to be combined to create innovative connected products and services.

Acknowledgements

This research has been supported by the Federal Ministry of Economic Affairs and Energy of Germany for the project GeniusTex as part of the strategic projects for "Smart Service Welt". The consortium consists of Fraunhofer FIT, Infineon Technologies AG, ASYS Group, ottobock GmbH, Technology and Innovation Management Group at RWTH Aachen and is lead by Institute of Textile Technology (ITA) at RWTH Aachen University.

References

[1] I. Gehrke, V. Tenner, V. Lutz, D. Schmelzeisen, and T. Gries, *Smart Textiles Production: Overview of Materials, Sensor and Production Technologies for Industrial Smart Textiles*. MDPI, 2019.

[2] CEN European Committee for Standardization, "Textilien und textile produkte - intelligente textilien - definitionen, klassifizierung, anwendungen und normungsbedarf," Berlin, 2012.

[3] O. Döhne, "Endlich wieder oberwasser: Technische und smarte textilien," Markets International, Berlin, 2018.

[4] T. Kirstein, Ed., *Multidisciplinary know-how for smart textiles developers*, ser. Woodhead publishing series in textiles, 2042-0803. Oxford and Philadelphia: Woodhead Pub, 2013, vol. no. 139.

[5] A. Lymberis, "Smart wearables reflection and orientation paper: Digital industry competitive electronics industry," 2017.

[6] A. Garlinska and A. Röpert, "Technology management and innovation strategies in the development of smart textiles," in *Multidisciplinary Know-How for Smart-Textiles Developers*. Elsevier, 2013, pp. 369–398.

[7] I. Gehrke, M. Knuth, and S. Tramp, "Geniustex - smart textiles ontology," 2019. [Online]. Available: https://vocab.eccenca.com/geniustex/

[8] A. Griffin, "The effect of project and process characteristics on product development cycle time," *Journal of Marketing Research*, vol. 34, no. 1, pp. 24–35, 1997.

[9] J. Göpfert, "Modularisierung von technik und organisation in der produktentwicklung," in *Modulare Produktentwicklung*. Springer, 1998, pp. 59–203.

[10] A. Duque, C. Campos, E. Jiménez-Ruiz, and R. Chalmeta, "An ontological solution to support interoperability in the textile industry," in *IFIP-International Workshop on Enterprise Interoperability*. Springer, 2009, pp. 38–51.

[11] A. L. Szejka, O. C. Júnior, E. R. Loures, H. Panetto, and A. Aubry, "Proposal of a model-driven ontology for product development process interoperability and information sharing," in *Product Lifecycle Management for Digital Transformation of Industries*, R. Harik, L. Rivest, A. Bernard, B. Eynard, and A. Bouras, Eds. Cham: Springer International Publishing, 2016, pp. 158–168.

[12] Z. Y. Wu, X. G. Ming, L. N. He, M. Li, and X. Z. Li, "Knowledge integration and sharing for complex product development," *International Journal of Production Research*, vol. 52, no. 21, pp. 6296–6313, 2014.

[13] T. R. Gruber, "A translation approach to portable ontology specifications," *Knowledge acquisition*, vol. 5, no. 2, pp. 199–220, 1993.

[14] A. Petrovan, M. Lobontiu, G. Lobontiu, and S. R. Nagy, "Overview on equipment development ontology," *Applied Mechanics and Materials*, vol. 657, pp. 1066–1070, 2014.

[15] T. Liu, H. Wang, and Y. He, "Intelligent knowledge recommending approach for new product development based on workflow context matching," *Concurrent Engineering*, vol. 24, no. 4, pp. 318–329, 2016.

[16] O. Matei and D. Contras, "Automated product design and development using evolutionary ontology," in *Artificial Intelligence Perspectives in Intelligent Systems*. Springer, 2016, pp. 47–57.

[17] W. Song, X. Ming, and P. Wang, "Collaborative product innovation network: Status review, framework, and technology solutions," *Concurrent Engineering*, vol. 21, no. 1, pp. 55–64, 2013.

 ISBN: 978-1-61208-775-7

[18] X. Li, Z. Wu, M. Goh, and S. Qiu, "Ontological knowledge integration and sharing for collaborative product development," *International Journal of Computer Integrated Manufacturing*, vol. 31, no. 3, pp. 275–288, 2018.

[19] P. de Sabbata, N. Gessa, G. D'Agosta, M. Busanelli, C. Novelli, and G.-A. Kartsounis, "Knowledge exchange infrastructure to support extended smart garment organizations," *Transforming Clothing Production into a Demand Driven, Knowledgebased High Tech Industry, the Leapfrog Paradigm*, 2009.

[20] C. P. Ferrero, E. Lloret, and M. Palomar, "Towards the design of a textile chemical ontology," *International Journal of Advanced Computer Science and Applications, Special Issue on Natural Language Processing*, pp. 35–41, 2014.

[21] X. Aimé, S. Georges, and J. Hornung, "Vetivoc: a modular ontology for the fashion, textile, and clothing domain," *Applied ontology*, 10 2015.

[22] J. Kim, J. Kim, D. Lee, and K.-Y. Chung, "Ontology driven interactive healthcare with wearable sensors," *Multimedia Tools and Applications*, vol. 71, no. 2, pp. 827–841, 2014.

[23] L. Xin, W. Xue-fen, and Z. Tao, Eds., *An e-Textile human gesture ontology model: 2011 4th International Congress on Image and Signal Processing*, vol. 3, 2011.

[24] G. Pioggia, G. Ricci, S. Bonfiglio, E. Bekiaris, G. Siciliano, and D. De Rossi, Eds., *An Ontology-Driven Multisensorial Platform to Enable Unobtrusive Human Monitoring and Independent Living: 2009 Ninth International Conference on Intelligent Systems Design and Applications*, 2009.

[25] A. Schwarz, L. van Langenhove, P. Guermonprez, and D. Deguillemont, "A roadmap on smart textiles," *Textile Progress*, vol. 42, no. 2, pp. 99–180, 2010.

[26] J. Cheng, B. Zhou, P. Lukowicz, F. Seoane, M. Varga, A. Mehmann, P. Chabrecek, W. Gaschler, K. Goenner, and H. Horter, "Textile building blocks: Toward simple, modularized, and standardized smart textile," in *Smart Textiles*, S. Schneegass, Ed. Springer, 2017, pp. 303–331.

[27] D. Stricker, "Easy-imp collaborative development of intelligent wearable meta-products in the cloud: Schlussbericht; förderkennzeichen fp7-nmp 609078."

[28] D. Garijo, "Widoco: A wizard for documenting ontologies." in *International Semantic Web Conference (2)*, ser. Lecture Notes in Computer Science, C. d'Amato, M. Fernndez, V. A. M. Tamma, F. Lcu, P. Cudr-Mauroux, J. F. Sequeda, C. Lange, and J. Heflin, Eds., vol. 10588. Springer, 2017, pp. 94–102.

[29] R. Atkinson, R. Garca-Castro, J. Lieberman, and C. Stadler, "Semantic sensor network ontology," 2017. [Online]. Available: https://www.w3.org/TR/vocab-ssn/

[30] H. Rijgersberg, "Om - ontology of units of measure," 2013. [Online]. Available: https://github.com/HajoRijgersberg/OM

[31] "Pellet: An open source owl dl reasoner for java," 2019. [Online]. Available: https://github.com/stardog-union/pellet

[32] D. Kontokostas, P. Westphal, S. Auer, S. Hellmann, J. Lehmann, R. Cornelissen, and A. Zaveri, "Test-driven evaluation of linked data quality," in *Proceedings of the 23rd International Conference on World Wide Web*, ser. WWW '14. International World Wide Web Conferences Steering Committee, 2014, pp. 747–758.

[33] M. Lefranois and A. Zimmermann, "Supporting arbitrary custom datatypes in rdf and sparql." in *ESWC*, ser. Lecture Notes in Computer Science, H. Sack, E. Blomqvist, M. d'Aquin, C. Ghidini, S. P. Ponzetto, and C. Lange, Eds., vol. 9678. Springer, 2016, pp. 371–386.

[34] J. M. Keil and S. Schindler, "Comparison and evaluation of ontologies for units of measurement." *Semantic Web*, vol. 10, no. 1, pp. 33–51, 2019.

 ISBN: 978-1-61208-775-7

Synonym Predicate Discovery for Linked Data Quality Assessment Without Requiring the Ontology Semantic Relations

Samah Salem
Lire Laboratory, dept. of TLSI
Abdelhamid Mehri- Constantine 2 University
Constantine, Algeria
E-mail: samah.salem@univ-constantine2.dz

Fouzia Benchikha
Lire Laboratory, dept. of TLSI
Abdelhamid Mehri- Constantine 2 University
Constantine, Algeria
E-mail: fouzia.benchikha@ univ-constantine2.dz

***Abstract*—Over the past years, an increasing number of datasets have been published as part of the Web of Data, reaching more than 1,200 datasets in 2019. However, many datasets, totaling a large quantity of RDF triples, are without ontology or with an incomplete one. As a result, they suffer more and more from quality problems. Assessing linked data quality for fitness for use is a current research problem that we are interested in. In this paper, we propose a novel approach for the assessment of quality between RDF triples without requiring schema information. It allows assessing the quality of datasets by detecting errors and eventually measuring the error rate using synonym predicates techniques, profiling statistics, and quality verification cases. Promising results are obtained on the DBpedia dataset where several data quality issues have been detected, such as inaccurate values, redundant predicates, and redundant triples.**

Keywords-linked data; quality assessment; semantic relations; synonym predicates; profiling statistics; DBpedia.

I. INTRODUCTION

In the last decade, the number of datasets published in the Linked Data (LD) format had increased from 12 datasets in May 2007 to 1,239 datasets in March 2019. This huge growth leads to the appearance of many structured datasets on the Web of Data [18], such as DBpedia [19] and Wikidata [20]. However, many of these datasets do not have a well-developed ontology or do not have an ontology at all, and their qualities are highly variable, as in the case of DBpedia that is considered as the most well organized and widely used LD resource [2].

In the literature, data quality is usually defined as "fitness for use". It depends on several dimensions, such as accuracy, completeness, relevance, credibility, comprehensibility, consistency, and conciseness [1]. Several authors have proposed interesting approaches for quality assessment requiring ontology for datasets, which is not always available or may be incomplete. New approaches are thus required to deal with LD quality assessment by finding features that best represent the semantics of Resource Description Framework (RDF) triples without requiring ontologies, when each triple represents two entities (Subject and Object) linked with (Predicate). To achieve this goal, we propose an approach for quality assessment between RDF triples, independently of the semantic relationships of the ontology, using both techniques of synonym predicates discovery, profiling statistics, and predefined quality verification cases.

The remainder of this paper is structured as follows: in Section II, we discuss the related work. Section III presents our proposed approach. An evaluation is given in Section IV. Finally, we conclude with ideas for future work in Section V.

II. RELATED WORK

In this section, we present related work on quality assessment in the web of data as well as existing approaches for synonym predicates discovery.

A. Linked Data Quality Assessment

Several works on the quality assessment of linked data have been proposed. They focused on assessing the quality of different parts of datasets, namely literals, predicates, triples, and metadata. We present here the well-known methodologies and tools, which could be classified into two distinct categories: (1) those that use ontologies and (2) those that do not use ontologies.

In the first category, several approaches are proposed. Lei et al. [5] propose a framework that allows evaluating the accuracy, consistency, and conciseness of semantic metadata. SWIQA [4] allows automatically evaluating the quality of published data using a quality rule template. In addition, RDFUnit [3], a pattern-based approach for LD quality assessment, uses data schema and quality patterns are created from DBpedia user community feedback, Wikipedia maintenance system, and ontology analysis. Besides, another approach called ABSTAT [6], allows the use of data profiling and data mining techniques to explore LD and to detect quality issues at the schema level. Finally, a semi-automatic methodology for dataset quality assessment and improvement is proposed in [14]. Although, the previous works provide good support for LD quality assessment, none of them is focused on detecting errors by discovering semantic relations between properties in the dataset (that lacks a well-developed ontology or does not have ontology at all). Therefore, there is still a need for additional researches and efficient techniques to provide high quality for LD that do not require a lot of user expertise and ontology information.

 ISBN: 978-1-61208-775-7

In contrast to the first category, in the second one, the most significant research work consists of the approach proposed by Jang et al. [2], which assesses the LD quality without using any data schema. It measures the quality of LD in terms of property's domain, range and data type through a semi-automatic generation of data quality patterns. The approach has been applied to Korean DBpedia, in which an error occurrence rate equal to 36.31% has been obtained. It seems to be an interesting approach, which will open new possibilities for researchers to develop efficient techniques for LD quality assessment without using data schema information. However, the quality assessment is done with only one triple and it does not give the exact domain/range (i.e., the generation of an upper-class type). Moreover, no quality improvement after detecting quality problems is incorporated.

In the context of our work, we consider datasets without ontologies. We propose an approach for LD quality assessment by understanding semantics between properties and considering assessing quality between triples. Table I gives a comparative study.

The proposed approach is based on synonym predicates discovery to efficiently assess data quality through detecting errors between triples. The next subsection will present some existing techniques of synonym predicates discovery that has been used for different purposes.

B. Synonym Predicates Discovery

In the literature, some work use synonym predicates discovery techniques in LD. For instance, Abedjan and Naumann [8] propose an approach that allows discovering synonymously used predicates. The main objective is to expand queries, by aggregating positive and negative association rules at the statement level based on the concept of mining configurations. However, it discovers only predicates that could substitute each other, such as *starring* and *artist*, which is usually not suitable since the predicate expansion operation is different from the predicate unification operation.

Another work for knowledge graph consolidation is proposed in [9]. It is a data-driven method to identify existed synonymous relationships in the knowledge graph using knowledge embedding methods, such as RESCAL [11], ComlEX [12], and ANALOGY [13], and without making any assumptions on the data.

In addition, Issa [10] proposed an approach to assess the completeness and the conciseness of LD. It is based on Abedjan et al. [8] approach, in which synonymous relationships are used to detect redundant predicates in datasets and so to ensure their conciseness.

Broadly, in the existing approaches, the synonym predicates are used for query expansion [8], graph consolidation [9], and redundancy detection [10], but in our approach, we discover the synonym predicates for a holistic detection of quality issues at subject-level, predicate-level, and object-level. Since in our opinion, the discovery of synonyms may reveal several problems in the data. As well, the methods used for the discovery of synonyms are different from our natural language processing method. Table II highlights their main limitations compared with our approach. The next section will give more details on the proposed approach.

TABLE I. COMPARISON BETWEEN LINKED DATA QUALITY ASSESSMENT APPROACHES.

Approaches	Goal	Quality of	Quality dimensions	With/ without ontology
Lei et al., 2007	Quality assessment of semantic metadata	Metadata	Accuracy, consistency, conciseness	With ontology
Fürber and Hepp, 2011	Quality assessment of published data	Literal	Accuracy, completeness, uniqueness, timeliness	With ontology
Kontokostas et al., 2014	DBpedia quality assessment	Triple	-	With ontology
Spahiu et al., 2016	Summarize the content of a dataset and reveal data quality problems	Predicate	Accuracy, completeness, timeliness	With ontology
Jang et al., 2015	Linked data quality assessment	Triple	Accuracy and consistency	Without ontology
Our approach	Assess the quality between RDF triples Understand the semantics between properties	Predicate, object, triple	Accuracy and conciseness	Without ontology

TABLE II. COMPARISON BETWEEN OUR APPROACH AND SYNONYM PREDICATE DISCOVERY APPROACHES.

Approaches	Goal	Based on	Techniques
Abedjan and Naumann, 2013	Query expansion	Synonymously used predicates	Association rules mining
Issa, 2018	Dataset conciseness	Synonymously used predicates	Abedjan and Naumann. [8] approach
Kalo et al., 2019	Graph consolidation	Synonym predicates	Knowledge embedding
Our approach	Measure the accuracy and the conciseness of the datasets that do not have an ontology	Synonym predicates	Natural language processing-based methods

III. THE PROPOSED APPROACH

We propose a novel approach for the assessment of quality between RDF triples without requiring schema information. The approach consists of three main steps (as shown in Figure 1): (1) synonym predicates discovery, (2) profiling statistics generation, and (3) quality assessment. It assesses the quality of datasets by detecting errors and eventually measuring the error rate using synonym predicates techniques, profiling statistics, and quality verification cases.

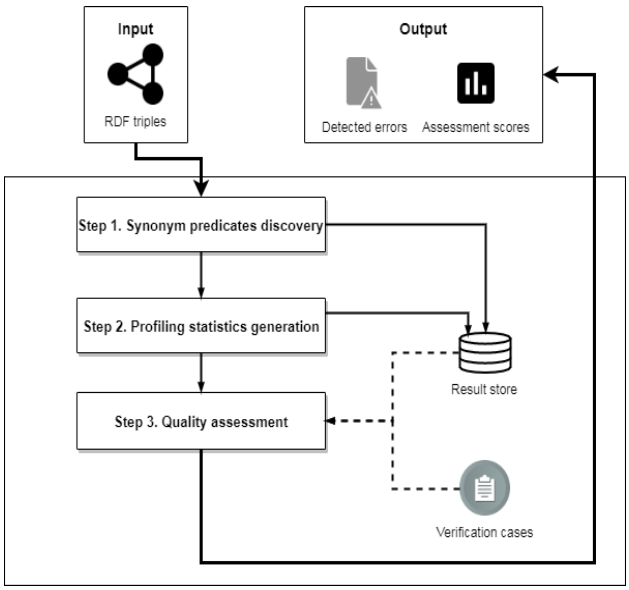

Figure 1. A three-step approach for quality assessment.

A. Step 1. Synonym Predicates Discovery

In a dataset without schema, there is no definition of entities, data types, and semantics of the properties. However, the possibility of finding two or more predicates, which have the same meaning is very high (as revealed after a study on DBpedia, for example *foaf:nick* and *dbp:nickname*). For this purpose, we are interested particularly in synonym predicates discovery for the creation of synonym-pattern (cf. III.B) and for the detection of quality problems (cf. III.C).

Our research on discovering the synonym predicates is based on the natural language processing methods. Indeed, as it is known that the web of data uses complex identifiers for naming predicates and not literals, then we adapt the natural language processing methods to our validation needs. An RDF graph G is a set of triples T s.t. G = {T}, where each triple T has the form of subject, predicate, and object s.t. T = (s,p,o).

$$\exists\, o_i, \exists\, s_i \mid p_i(o_i, s_i) \quad (1)$$

$$\exists\, o_j, \exists\, s_j \mid p_j(o_j, s_j) \quad (2)$$

As in our case, we are interested in the errors that occur between triples, therefore we will focus on the discovery of the synonym predicates. (3) Gives a predicate p_i of triple t_i and predicate p_j candidate synonym of triple t_j.

$$p_i \in t_i \wedge p_j \in t_j \mid p_i \equiv_{syn} p_j \quad (3)$$

We focus on the thesaurus-based methods, WordNet, due to their high precision in the synonym identification that is necessary in our case study. However, there are several synonyms that are not indexed by WordNet, and the problem of the predicates with spelling errors that are not detected by the WordNet, such as *dbp:birthPace*, *dbp:birthPaxes*, and *dbp:nbirthPlace*. For these reasons, we use a check spelling method [21] that suggests corrections for misspelt words based on many popular spells checking packages, such as Ispell [15], Aspell [16], and MySpell [17]. This is a semi-automatic step, and a user (not necessarily a domain expert) must confirm each detected synonym pair.

B. Step 2. Profiling Statistics Generation

Data profiling is about examining and collecting information from datasets. In our approach, it is very useful to generate some profiling tasks to prepare for quality score estimation. The principal goal of this step is to generate synonym-pattern based on the results of the first step, and to calculate simple profiling statistics, such as the total number of triples in a dataset, and the property occurrence (i.e., how many times the property defined as a synonym occurs in the dataset). The synonym-pattern is a summary that provides a global view of the synonym predicates existing in the dataset and the predicate frequency. A predicate-pattern has the following form:

$$<p_i\,(\Sigma p_i) \equiv_{syn} p_j(\Sigma p_j) \equiv_{syn} p_n(\Sigma p_n)>. \quad (4)$$

For example, we can have as result <dbo:birthplace (13), dbp:birthCity (2)>, where the pattern shows two predicates synonym (dbo:birthplace and dbp:birthCity) with the frequency of each predicate (13 and 2 respectively) in the dataset.

C. Step 3. Quality Assessment

In the previous steps, we generated the synonym predicates and the profiling statistics. This step involves the actual quality assessment including: (1) the detection of quality problems that may occur between RDF triples, and (2) the estimation of quality scores. For the first task, we will use the synonym predicates defined in the first step, and predefined quality verification cases (more details are provided below). For the second task, we will use the profiling statistics generated in the second step for the estimation of quality scores. Note that, in this first version of the proposed approach, we allow just to reveal the errors existing between RDF triples, in the future, we will incorporate the treatment of errors once identified.

 ISBN: 978-1-61208-775-7

1) Quality Problems Detection: In order to detect quality issues, we will verify the similarity or the difference between the subject and the object of each predicate synonyms pair to detect the errors between RDF triples. Note that there are only four possible cases that could occur between two triples.

a) Case 01:

$$\text{If } s_i = s_j \wedge o_i = o_j \Rightarrow \{p_i(o_i,s_i) \Leftrightarrow p_j(o_j,s_j)\}. \quad (5)$$

If the synonym predicates p_i and p_j have the same subject and the same object, then the triple t_i is equivalent to the triple t_j, which mean that one of these triple t_i or t_j is a redundant one (see TABLE IV).

b) Case 02:

$$\text{If } s_i = s_j \wedge o_i \neq o_j \Rightarrow \{p_i \Leftrightarrow p_j\}. \quad (6)$$

If the synonym predicates p_i and p_j have the same subject and totally different object, then (see TABLE IV), there are two types of errors:

- The predicate p_i is equivalent to the predicate p_j, which means that two predicates having the same meaning are defined differently in graph G, thus duplicating the information (i.e., redundant terms to represent the same predicate).
- We can ensure that the object value o_i and/ or o_j is an inaccurate value.

c) Case 03:

$$\text{If } s_i \neq s_j \wedge o_i = o_j \Rightarrow \{p_i \Leftrightarrow p_j\}. \quad (7)$$

If the synonym predicates p_i and p_j have different subject and the same object, then, it is possible to find two types of errors (see TABLE IV):

- The predicates p_i and p_j are defined differently, despite that they had the same meaning, since their equivalence.
- We can *assume* that the object value o_i and/ or o_j is an inaccurate value. If the predicate must contain a unique object value, then, we can *ensure* that the object value o_i and/ or o_j is an inaccurate value.

d) Case 04:

$$\text{If } s_i \neq s_j \wedge o_i \neq o_j \Rightarrow \{p_i \Leftrightarrow p_j\}. \quad (8)$$

If the synonym predicates p_i and p_j have different subjects and different objects, then, we can say that in this case there is duplicate information in order to define the same predicate in the dataset (see TABLE IV).

2) Quality Scores Estimation: After detecting the abnormal triples, it is suitable to measure the quality in terms of numbers. Based on the data quality score metrics [4][14] and the generated profiling statistics, the quality scores according to our needs are calculated, in particularly quality score (QScore), accuracy (Acc-QS), and conciseness (Co-QS). For instance, QScore is the ratio between the number of abnormal triples A_t and the total number of triples T_t, as the following formula shows:

$$\text{QScore} = A_t / T_t. \quad (9)$$

In addition, in order to differentiate between the detected errors, we calculate Acc-QS to measure the percentage of inaccurate values, and Co-QS for duplicate predicates and triples.

$$\text{Acc-QS} = PA_t / A_t. \quad (10)$$

$$\text{Co-QS} = PC_t / A_t. \quad (11)$$

Where PA_t is the number of inaccurate values, and PC_t represents the number of redundant predicates plus the number of redundant triples. The obtained results present the accuracy/ conciseness error occurrence rate compared to the total number of errors in the dataset.

IV. VALIDATION

In order to evaluate our proposed approach, which is available on GitHub repository [22], several studies are carried out on the latest version of DBpedia released in 2019. The experiment revealed several cases of unknown synonymous relationships. Table III illustrates some synonym pairs discovered by applying our approach to entities of type *Person*. Quality problems between triples are detected as shown in Table IV. We used properties of 449 triples, and we found 50 abnormal triples that present an error rate equal to 11 %. In order to better evaluate the performance of the proposed approach, it will be applied to even larger and more complex datasets (which is left for future work).

TABLE III. TOP 5 OF SYNONYM PAIRS.

DBpedia Person	
foaf:name	dbp:name
dbo:birthplace	dbp:birthCity
dbo:birthDate	dbp:birthdate
foaf:gender	dbo:gender
dbo:occupation	dbp:occupation

The abnormal triples may contain several errors, such as redundant predicates, redundant triples, and inaccurate values. Through the detection of these errors, we could measure two quality dimensions, namely accuracy, and conciseness of the dataset. Note that we omit the blank node from our approach and leave it for future work.

 ISBN: 978-1-61208-775-7

TABLE I. QUALITY ISSUES DETECTED BETWEEN TRIPLES ON DBPEDIA.

Triples pairs with synonym predicates	Error type	Quality dimension
dbr:Duduka_da_Fonseca, **dbo:birthplace**, dbr:Rio_de_Janeiro dbr:Duduka_da_Fonseca, **dbp:birthCity**, dbr:Rio_de_Janeiro	***Case 01:*** The results show that the two triples are equivalent, which means that one of these two triples is redundant.	*Conciseness*
dbr:Paulie_Pennino, **foaf:gender**, "female"@en dbr:Paulie_Pennino, **dbo:gender**, dbr:Male	***Case 02:*** The sex of the entity **dbr:Paulie_Pennino** is inaccurate in one of these two triples since once is defined as "female", and once is defined as dbr:Male	*Accuracy/ Conciseness*
dbr:Cornelia_(wife_of_Caesar), **dbp:diedPlace**, dbr:Rome dbr:Aloysius_Lilius, **dbo:deathPlace**, dbr:Rome	***Case 03:*** The predicates **dbp:diedPlace** and **dbo:deathPlace** are defined differently despite that they have the same meaning	*Conciseness*
dbr:Alice_Walker, **foaf:gender**, "female"@en dbr:Zack_Addy, **dbo:gender**, dbr:Male	***Case 04:*** In this case, there is duplicate information in order to define the same predicate in the dataset	*Conciseness*

V. CONCLUSION

The Web of Data allows publishing data that includes its semantics using shared vocabularies and data annotations described in ontologies [4]. Unfortunately, there are a large number of datasets without ontology or with an incomplete one. Therefore, it is necessary to generate ontologies from the target LD. However, constructing an ontology for a large amount of data that may contain quality problems is a difficult and time-wasting task. For these reasons, we propose an approach based on the discovery of the semantic links between properties to assess the quality of RDF triples without requiring the existence of the ontology information. This work guides the users to evaluate the quality between RDF triples through the discovery of synonym predicates and the generation of profiling statistics, and predefined quality verification cases.

Similar to [9][10] approaches, we are interested in the discovery of the synonym predicates, but in our approach we work with RDF triples without using the ontology information. We present the discovered synonym predicates as a synonym-pattern in order to (i) understand the semantics between properties, (ii) detect quality problems and (iii) estimate the quality scores. Our approach allows to efficiently detecting the errors between RDF triples without using the ontology information at all. The obtained results show that there is an important number of inaccurate values in the DBpedia dataset, as well as, duplicate predicates due to the usage of synonym predicates discovery. Despite the fact that the proposed approach shows interesting results in the field of quality problem detection, some exceptions will be handled in the future. For example, when the predicate values are represented with different patterns, such as (dbr:Julius_Caesar, dbo:birthdate, '−100 - 07 - 13') and (dbr:Julius_Caesar, dbo:birthdate, '− 100 - 7 - 13') these triples are identified in Case 02, however, they should be identified in Case 01.

For further work, we intend to define more varied metrics for linked data quality assessment mainly for dataset without ontology. We plan to improve the quality of data and to improve the performance of our approach through the treatment of the blank node identifiers.

REFERENCES

[1] A. Zaverii et al., "Quality assessment for linked data: A survey," Semantic Web, 7(1), pp. 63-93, january 2016.

[2] S. Jang, M. Megawati, J. Choi, and M. Y. Yi, "Semi-automatic quality assessment of linked data without requiring ontology," In NLP-DBPEDIA@ ISWC, pp. 45-55, October 2015.

[3] D. Kontokostas et al., "Test-driven evaluation of linked data quality," In Proceedings of the 23rd international conference on World Wide Web, pp. 747-758, ACM, April 2014.

[4] C. Fürber and M. Hepp, "Swiqa-a semantic web information quality assessment framework," In ECIS, Vol. 15, pp. 19-31, 2011.

[5] Y. Lei, V. Uren, and E. Motta, "A framework for evaluating semantic metadata," Proceedings of the 4th international conference on Knowledge capture, ACM, pp. 135-142, October 2007.

[6] B. Spahiu, "Profiling the Linked (Open) Data," International Semantic Web Conference, Vol. 1491, October 2015.

[7] B. Spahiu, R. Porrini, M. Palmonari, A. Rula, and A. Maurino, "ABSTAT: ontology-driven linked data summaries with pattern minimalization," In European Semantic Web Conference, pp. 381-395. Springer, Cham, 2016.

[8] Z. Abedjan and F. Naumann, "Synonym analysis for predicate expansion." In Extended semantic web conference, pp. 140-154. Springer, Berlin, Heidelberg, May 2013.

[9] C. Kalo, P. Ehler, and W. T. Balke, "Knowledge Graph Consolidation by Unifying Synonymous Relationships," International Semantic Web Conference. Springer, pp. 276-292, Cham, October 2019.

[10] S. Issa, "Linked Data Quality," In DC@ ISWC, pp. 37-45, 2018.

[11] M. Nickel, V. Tresp, and H. P. Kriegel, "A Three-Way Model for Collective Learning on Multi-Relational Data," In ICML, vol. 11, pp. 809-816, June 2011.

[12] T. Trouillon, J. Welbl, S. Riedel, E. Gaussier, and G. Bouchard, "Complex embeddings for simple link prediction," In International Conference on Machine Learning, pp. 2071-2080, June 2016.

[13] H. Liu, Y. Wu, and Y. Yang, " Analogical inference for multi-relational embeddings." In Proceedings of the 34th International Conference on Machine Learning-Volume 70, pp. 2168-2178, JMLR. org, August 2017.

[14] A. Rula and A. Zaveri, "Methodology for Assessment of Linked Data Quality," In Proceedings of the 1st Workshop on Linked Data Quality co-located with 10th International

Conference on Semantic Systems, LDQ@SEMANTiCS 2014, Leipzig, Germany, September 2nd, 2014.

[15] R. E. Gorin, P. Willisson, W. Buehring, and G. Kuenning, "Ispell, a free software package for spell checking files," The UNIX community, 1971.

[16] K. Atkinson, "GNU Aspell," 2003, URL http://aspell. Net, 2011.

[17] C. Andrea, "My spell-checker's «weigh» with words," The Christian Science Monitor, August 2002.

[18] The Linked Open Data Cloud. [Online]. Available from: https://lod-cloud.net/, last accessed: December 19,2019.

[19] DBpedia. [Online]. Available from: https://wiki.dbpedia.org/, last accessed: December 19,2019.

[20] Wikidata. [Online]. Available from: https://www.wikidata.org/wiki/Wikidata:Main_Page, last accessed: December 19,2019.

[21] Spellchecking library for Python. [Online]. Available from: https://github.com/pyenchant/pyenchant, last accessed: December 19,2019.

[22] https://github.com/SalemSamah/SPDiscovery.

 ISBN: 978-1-61208-775-7

Statistical Analysis of Stock Profits to Evaluate Performance of Markets

Yoshihisa Udagawa
Faculty of Informatics, Tokyo University of Information Sciences
Chiba-city, Chiba, Japan
e-mail: yu207233@rsch.tuis.ac.jp

***Abstract*— Candlestick charting is one of the most popular techniques used to predict short-term stock price trends. Despite popularity, there is still no consistent conclusion for the predictability of the technique mainly due to qualitative description of candlestick patterns. This paper proposes a retrieval model with six parameters that allows us to define both candlestick patterns and price zones where the patterns occur. Because criteria that trigger exit from a market largely affect profits and losses, we propose three market exit criteria. Simulations to estimate profits are performed using five global markets with approximately the same parameters for the retrieval model and the market exit criteria. The results of simulations indicate that the proposed method leads to trades with around 85% of successful stock trades in the case of a typical uptrend candlestick pattern. Five global markets are also analyzed and compared to show graphically the profitability of the markets based on simulated profits.**

Keywords*— *Stock price prediction; Technical analysis; Candlestick charts; Market exit criteria; Profit simulation; Global market comparison.

I. INTRODUCTION

Forecasting a direction of future stock prices attracts the attention of not only financial investors but also researchers in computer science. The common motivation is to predict the future direction of prices for successful stock trade and developing computer system to support a trader. While many researches on stock price prediction focus on a specific market, some researches deal with multiple stock markets to seek global investment opportunities.

Dimson et al. [1] discuss performances of global markets including emerging and developed ones. Though emerging markets have grown to a significant size up to 2007, developed markets, notably US markets, have outpaced the growth in emerging markets in the 21st century because of global financial crisis. Ahmad et al. [2] measure the impact of volatility in six emerging stock markets of Asia. The results of statistical analyses show volatility is significantly related to return in each market.

What is missing in the research of Dimson et al. [1] is the lack of algorithmic and statistical analyses for objective comparison among global markets in terms of profits from a trader's point of view. They discuss performances of global markets from an economic point of view without mentioning results of statistical analyses. They suggest that US stock markets are more profitable than others and that is the same conclusion to which we arrived in this paper. Meanwhile, Ahmad et al. [2] examine volatilities among Asian stock markets to find out a causal relation between volatility and stock returns. While their research uses statistical analyses, it fails to include developed markets.

The purpose of this resarch is to compare profitability of emerging and developed markets based on algorithmic and statistical analyses. We develop a simulator program in Java that implements a retrieval model to find opportunities for buying stocks, and algorithms to trigger selling stocks to lead to profitable trade. The experimental results are statistically analyzed to examine the extent of relationship among measured variables. Simulated profits are displayed in bar graphs to easily compare the global stock markets under discussion.

The contributions of this paper are as follows:

I. Proposal of a model using six parameters to retrieve candlestick patterns that are both similar in price patterns and price level, i.e., price high and/or low zone in which they occur.
II. Proposal of three novel algorithms to trigger selling stocks to fix profit in case of a long market position.
III. Evaluation of performances of the proposed model and the algorithms to trigger selling stocks through simulations in terms of profit.

The remainder of the paper is organized as follows. Section II recapitulates some related work. Section III gives backgrounds of the candlestick charting. Section IV proposes a model for retrieving similar candlestick charts and the triggering algorithms. Section V presents experimental results on a strong uptrend pattern using five markets' data in US and Asia stock markets. Section VI concludes the paper with our plans for future work.

II. RELATED WORK

There have been a growing number of studies on predicting future movements of stock markets. In this section, we review previous studies on performances of global markets and predictabilities of candlestick patterns.

A. Studies on Performances of Global Markets

Dimson et al. [1] discuss performances of markets in emerging and developed countries from an economic perspective. They find that emerging markets achieved a higher return of 11.7% per year than a developed markets' return of 10.5% from 1950 to 2019. However, because of the global financial crisis, the average return of US stocks is 10.6%, while that of the world stocks excluding US ones is

 ISBN: 978-1-61208-775-7

5.3% in the 21st century. They conclude that investors should be modest to invest in emerging markets because exchange rate movements are largely affected by inflation in emerging countries in addition to questionable capabilities to maintain a fair market.

Ahmad et al. [2] statistically examine six emerging Asian stock markets with respect to stock returns and volatility. The markets include KSE100 (Pakistan), Nikkei 225 (Japan), KOSPI (South Korea), Hang Seng (Hong Kong), SSE (China), and BSE (India). The results show that KOSPI has the highest average annual return of 12.67%, followed by BSE with 11.61%.

B. Studies on candlestick patterns

The researches in [4]-[13] are on usefulness of candlestick patterns in technical analysis [3]. Most researches focus on one market. Some researches use stock data of multiple markets, but their aims are to confirm their estimated profits.

1) Studies disapproving of candlestick patterns

As for the candlestick pattern method in technical analysis [3], several studies [4]-[6] conclude that it is useless based on the experiments using the stock exchange markets' data in the US, Japan and Thailand.

Horton [4] studies the profitability of 4 pairs of three-day candlestick patterns on 349 stocks that are representing major industry groups. The main conclusion of his study is that these candlestick patterns create no value for trading individual stocks.

Marshall et al. [5] find that under fixed holding period of 10 days, candlestick charting strategies are unprofitable for Dow Jones Industrial's components from 1992 to 2002. They also confirm that candlestick strategies generate no profit in Japanese markets from 1975 to 2004.

Based on experiments using stock data in the Stock Exchange of Thailand, Tharavanij et al. [6] conclude that any candlestick patterns cannot reliably predict market directions even with filtering by well-known stochastic oscillators [3].

2) Studies approving of candlestick patterns

Other studies conclude that applying certain candlestick patterns is profitable at least for short-term trading [7]-[13].

Caginalp et al. [7] study and favorably evaluate the predictive power of eight three-day reversal candlestick patterns on the S&P 500 index from 1992 to 1996. They propose to define candlestick patterns as a set of inequalities using opening, high, low, and closing prices. These inequalities are taken over in later studies.

Goo et al. [8] define 26 candlestick patterns using modified version of inequalities that are proposed by Caginalp et al. [7]. They examine these patterns using stock data of Taiwan markets, and conclude that the candlestick trading strategies are valuable for traders.

Chootong et al. [9] propose a trading strategy combining price movement patterns, candlestick chart patterns, and trading indicators. A neural network is employed to determine buy and sell signals. Experimental results using stock data in the Stock Exchange of Thailand show that the proposed strategy generally outperforms the use of traditional trading methods based on indicators.

One of the obstacles of candlestick charting is the highly subjective nature of candlestick pattern [3] since the candlestick patterns are defined using words and illustrations. Tsai et al. [10] propose an image processing technique to analyze the similarities of the candlestick charts. Their experimental results using Dow Jones Industrial Average index show that visual matching of candlestick charts is useful for predicting short-term stock movements.

Zhu et al. [11] examine the effectiveness of five different candlestick reversal patterns in predicting short-term stock movements. They use Chinese exchanges' data from 1999 to 2008 in the experiments. The results of statistical analyses suggest that the candlestick patterns perform well in predicting price trends.

Jamaloodeen et al. [12] statistically analyze the predictive power of two popular Japanese candlestick patterns, i.e., Shooting Star and Hammer patterns. They use over six decades of historical daily data of the S&P 500 index. Their findings include the two patterns are highly reliable when using high price for the Shooting Star and low price for the Hammer.

Udagawa [13] proposes a dynamic programing method to skip small and noisy candlesticks to improve predictability of candlestick charting. Experimental results show that the proposed method is effective in predicting both an uptrend and a downtrend.

The researches in [4]-[13] are dedicated to discuss effectiveness of candlestick patterns to spot a good opportunity for successful stock trading. On the other hand, this research aims to objectively compare profitability of emerging and developed markets using algorithms and statistical analyses. The proposed method implements a retrieval model that uniquely considers price zones in which candlestick patterns occur. In addition, the proposed method realizes algorithms that tell us when to sell stocks to fix profits, which allows us to estimate precise profits in stock trading, and to compare global markets in an objective manner.

III. CANDLESTICK CHART AND PATTERNS

This section introduces the formation of a candlestick. A series of candlesticks forms a candlestick pattern. Samples of well-known candlestick chart patterns that are believed to be useful for a successful investment are depicted. Criticism of candlestick patterns for predicting stock price movements are also mentioned.

A. Formation of Candlestick

A daily candlestick line is formed with the market's opening, high, low, and closing prices of a specific trading day [3]. The candlestick has a wide part, which is called *real body* representing the range between the opening and closing prices of that day's trading as shown in Figure 1. The color of the real body represents whether the opening price or the closing price is higher. If the price rises, a hollow body is drawn suggesting *bullish* or buying pressure. Otherwise a filled body is drawn suggesting *bearish* or selling pressure.

 ISBN: 978-1-61208-775-7

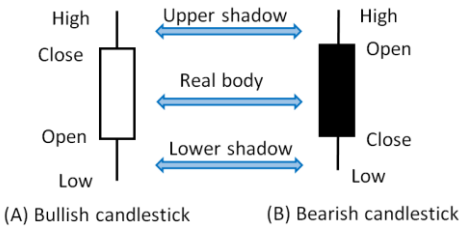

Figure 1. Candlestick formation

The thin lines above and below the body, which are named *shadows*, represent the range of prices traded in a day. The high is marked by the top of the upper shadow and the low by the bottom of the lower shadow.

B. Samples of Candlestick Patterns

Dozens of candlestick patterns are identified and become popular among worldwide stock traders [3]. These patterns have colorful names like *morning star*, *evening star*, *three white soldiers*, and *three crows*.

Figure 2 shows the *morning star* pattern which is considered as a major reversal signal when it appears in a price low zone or at a bottom. It consists of three candles, i.e., one short-bodied candle (filled or hollow) between a preceding long filled candle and a succeeding long hollow one. The pattern shows that the selling pressure that was there the day before is now subsiding. The third hollow candle overlaps with the body of the filled candle suggests a start of a bullish reversal. The larger the filled and hollow candles, and the higher the hollow candle moves, the larger the potential reversal. The opposite version of the *morning star* pattern is known as the *evening star* pattern which is a reversal signal when it appears in a price high zone or at the end of an uptrend.

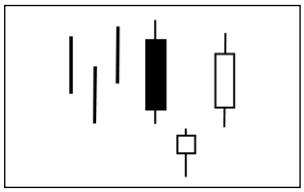

Figure 2. *Morning star* pattern

Figure 3 shows the *three white soldiers* pattern which is interpreted as a strong indication of a bullish market reversal when it appears in a price low zone.

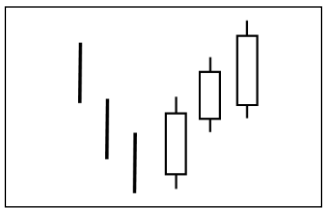

Figure 3. *Three white soldiers* pattern

It consists of three long hollow candles that close progressively higher on each subsequent trading day. Each candle opens higher than the previous opening price and closes near the high price of the day, showing a steady advance of buying pressure.

C. Criticism of Candlestick Patterns

The major criticism of the candlestick chart patterns is that the patterns are qualitatively described with words, such as "long/short candlesticks," "higher/lower trading," "strong/weak signal," supported by some illustrations [3]. Without modeling the candlestick patterns in a way that a computer can process and perform experiments comprehensively, arguments on the effectiveness of chart patterns would not come to an end.

In addition, some candlestick chart patterns yield a different, even opposite, forecast depending on whether they appear in price high and/or low zones. Formulating a suitable mathematical formulation of trend is still an open issue.

It deems that because of the lack of the strict definition of the candlestick chart patterns, mixed results are obtained in the studies on candlestick patterns. Negative conclusions to the predictability of candlesticks are reported [4]-[6], while positive evidences are provided for several candlestick chart patterns in experiments using U.S., Brazil and Asian stock markets [7]-[13].

IV. Proposed Model for Retrieving Candlestick Patterns

This section describes a model that allows us to retrieve similar to both candlestick patterns and price zones where the patterns occur. Three criteria that cope with moderate and sudden stock price changes are proposed to find opportunities for selling stocks.

A. Retrieval Model of Candlestick Patterns

After trial and error, we propose a model for retrieving similar candlestick charts that take into account where the stock price occurs in price zones in addition to a price change and a length of candlestick body. Figure 4 illustrates the model that consists of the six parameters as follows:

(1) Change of prices w.r.t previous closing price,
(2) Length of candlestick body,
(3) Difference between stock price and 5-day moving average,
(4) Difference between stock price and 25-day moving average,
(5) Slope of 5-day moving average,
(6) Slope of 25-day moving average.

 ISBN: 978-1-61208-775-7

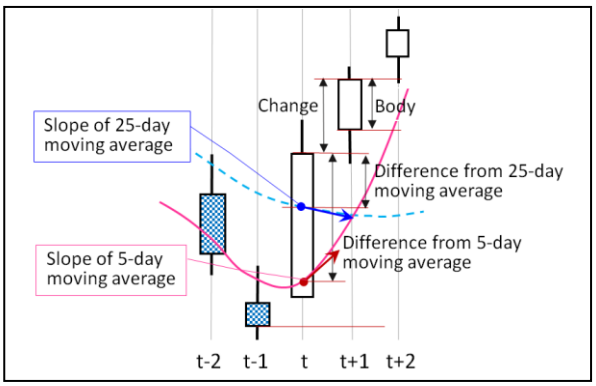

Figure 4. Candlestick pattern retrieval model

While most researches of candlestick patterns use a series of inequalities or technical indicators to identify stock price trends, i.e., an uptrend or a downtrend or a sideway (flat), the proposed model is unique in a sense that it uses two moving averages and their slopes. 5-day and 25-day moving averages are used since they are widely used in Japan. The moving averages are significant to identify the price zone where the candlestick pattern occurs. The slopes of the averages are also important to identify their trends.

Retrieval of similar candlestick charts in this research takes the following steps:

(1) Specify a reference day, i.e., typically a day of trend reversal, such as the last day of the *morning star* pattern in Figure 2.

(2) Define tolerances of the six parameters with respect to the reference day.

(3) Retrieve candidate candlesticks that satisfy all conditions C_1 to C_6.

C_1: if a difference between a closing price change of the reference day and that of a candidate day is within the change tolerance (*change_tol*), then C_1 is true.

C_2: if a difference between a body length of the reference day and that of a candidate day is within the body tolerance (*body_tol*), then C_2 is true.

C_3: if a difference between a closing price and a 5-day moving average of the reference day and that of a candidate day is within the tolerance (*av5diff_tol*), then C_3 is true.

C_4: if a difference between a closing price and a 25-day moving average of the reference day and that of a candidate day is within the tolerance (*av25diff_tol*), then C_4 is true.

C_5: if a slope of a 5-day moving average of the reference day and that of a candidate day is within the given tolerance (*slope5_tol*), then C_5 is true.

C_6: if a slope of a 25-day moving average of the reference day and that of a candidate day is within the given tolerance (*slope25_tol*), then C_6 is true.

(4) Check the conditions below on the two days following the reference day, i.e., ones labeled *t+1* and *t+2* in Figure 4.

F_1: if the change of the reference day and that of the day labeled *t+1*, i.e. the day after the reference day, are in the same direction.

F_2: if the change of the reference day and that of the day labeled *t+2*, are in the same direction.

Retrieval conditions of F_1 and F_2 are empirically derived. Setting values of six parameters are statistically determined to retrieve a suitable set of similar charts in order to analyze expected profits as described in Section V.

B. Finding selling opportunities

A set of similar candlesticks is retrieved by specifying a reference date and tolerances concerning six parameters shown in Figure 4. In the rest of the paper, we deal with patterns of uptrend in a long market position.

Traders will make a profit by a "buy low and sell high" strategy in uptrend. Candlestick patterns can suggest us when a specified trend begins, but do not cope with when a reverse of the trend begins. So, we need criteria or algorithms that tell us when to sell back stocks to fix profits and/or losses of a stock trading. One is the use of reversal candlestick patterns indicating downtrends. However, considering that candlestick patterns are derived from experience, use of the reversal patterns to trigger selling stocks seems incomplete and unprofitable due to dynamic nature of stock price movements.

In this study, we examine an approach using algorithm. Specifically, we decide an opportunity of selling stocks using the following three criteria:

(1) Sum of the negative change prices (*SumNC*) criterion: The value of *SumNC* is calculated by summing change prices in percentage that moved downward from the previous market day thorough a holding period. If The value exceeds a specified value then selling stocks is triggered.

(2) Sum of the negative differences from 5-day average (*SumND5av*) criterion:
The value of *SumND5av* is calculated by summing negative differences between a 5-day average and stock price thorough a holding period. Selling stocks is triggered when the value exceeds a specified limit.

(3) Plunge detection criterion:
This criterion intends to cope with quick price decline. When the stock price falls below the 5-day average then the range of price movements over the past 5 days (*PM5day*) is calculated. If the range is broader than a certain multiple of the standard deviation of price changes, the price fall is judged as a plunge and selling stocks is triggered.

V. Experimental Results

After outlining processes of experiments, statistical analyses of profits using Dow Jones Industrial Average, NASDAQ Composite index, Shanghai Composite index, Hang Seng index, and Nikkei Stock Average are discussed to evaluate performance of each market.

 ISBN: 978-1-61208-775-7

A. Data Conversion

The stock prices are converted to the ratio of closing prices. The conversion contributes to reduction of the effects of highness or lowness of the stock prices. The formula below is used for calculating the ratio of prices in a percentage.

$$R_i = (CP_i - CP_{i-1}) * 100 / CP_i \ (1 \leqq i \leqq n) \quad (1)$$

CP_i indicates the closing price of the i-th business date. CP_n means the closing price of the current date. R_n is the ratio of the difference between CP_n and CP_{n-1} to the closing price of the current date CP_n. The daily stock data from Nov. 25, 2009 to Dec. 24, 2019 are used in experiments on the research. The number of data is approximately 2,536 for each market.

B. Statistics of Candlestick Parameters

Table I summarizes statistics of six parameters concerning the proposed retrieval model of a candlestick pattern shown in Figure 4. The statistics of each parameter are calculated for all market days in the five markets. They provide basis of setting parameter values in this study.

TABLE I. SUMMARY OF STATISTICS OF SIX PARAMETERS

	Dow		NASDAQ		Nikkei 225		Shanghai		Hang Seng	
	Average	Deviation	Average	Deviation	Average	Deviation	Average	Deviation	Average	Deviation
Body length	0.0274	0.8433	0.0176	0.8670	−0.0082	0.9092	0.0981	1.2283	−0.0547	0.8024
Change	0.0435	0.8847	0.0623	1.0721	0.0454	1.2951	0.0053	1.3576	0.0149	1.1411
Difference of price and 5−day average	0.0720	0.9467	0.1005	1.1560	0.0583	1.4048	−0.0248	1.5335	0.0063	1.2700
Difference of price and 25−day average	0.4349	2.1902	0.5931	2.6560	0.3167	3.3890	−0.1672	4.1897	0.0357	3.1771
Slope of 5−day average	0.0390	0.3793	0.0548	0.4614	0.0359	0.5615	−0.0047	0.6275	0.0082	0.5167
Slope of 25−day average	0.0393	0.1496	0.0542	0.1821	0.0342	0.2370	−0.0042	0.2983	0.0085	0.2197

Averages of all six parameters are positive for Dow Jones and NASDAQ indicating that the two markets are on uptrends as a whole. Nikkei 225 and Hang Seng mark negative values for candlestick body length. In Shanghai, four parameters excluding candlestick body length and price change are negative values, suggesting that the market is less profitable than other markets.

C. Finding Profitable Trading Days for Each Market

Stock prices fluctuate depending not only on international but also domestic political and economic news. Therefore, the day suitable for buying stocks differs in each market. For a fair comparison of the markets, a preferred reference day for each market is determined by the following processes.

(1) For all market days, profit and/or loss of buying stock in a long position is calculated using a simulator.

(2) Sort market days by calculated profits, and select the day that generates the highest profit in 2019.

The following days are chosen as reference days.

- June 4, 2019: for DOW and NASDAC markets
- October 10, 2019: for Nikkei225 market
- January 4, 2019: for Hang Seng and Shanghai markets

Figure 5 shows the candlestick chart of Dow Jones index around June 4, 2019 coded by 0604.

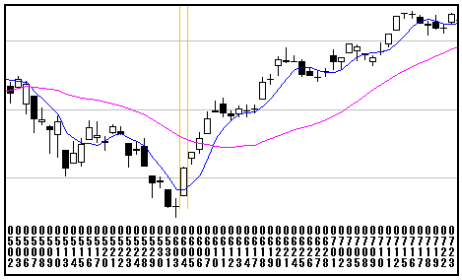

Figure 5. Candlestick chart of Dow Jones index around June 4, 2019

The day is the last day of a *morning star* pattern and the first day of a *three white soldiers* pattern. These patterns are known as strong uptrend patterns in the candlestick charting.

D. Experiments using Dow Jones Industrial Average

Experiments are performed using the GUI shown in Figure 6. Values of parameters used in the experiments are statistically determined.

First click on the *File* button to choose a CSV file containing a set of stock price data. The full path of the file is displayed. In Figure 6, a file named *Dow_ed.csv* is chosen.

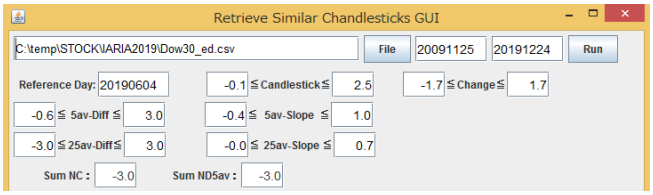

Figure 6. GUI for candlestick pattern retrieval model

The two text boxes in top right corner show periods of market days, i.e., 20091125 to 20191224. The text box labeled *Reference Day* specifies a reference market day that has a typical candlestick pattern for an uptrend reversal.

The two text boxes labeled *Candlestick* specify the tolerances of the length of the candlestick of a reference market day.

The two text boxes labeled *5av-Diff* mean the tolerances of the difference between the stock price and that of a 5-day average in percentage. The text boxes labeled *Change*, *25av-Diff*, *5av-Slope*, and *25av-Slope* are defined analogously.

The two text boxes labeled *SumNC* and *SumND5av* specify parameters to trigger selling stocks. The value of *PM5day*, which does not appear in Figure 6, is calculated by the following formula:

$$PM5day = 1 - (SumND5av / 5) \quad (2)$$

The formula is derived from experience so that the days of holding a stock is almost comparable to those of *SumNC* and *SumND5av*.

 ISBN: 978-1-61208-775-7

TABLE II. ESTIMATED PROFITS ON JUNE 4, 2019 AS REFERENCE MARKET DAY

MDay	x	y	z	SumNC			SumND5av			PM5day		
				Date to sell	Holding Days	Profit	Date to sell	Holding Days	Profit	Date to sell	Holding Days	Profit
20100216	1.680	0.394	0.812	20100331	31	5.605	20100427	49	6.885	20100427	49	6.885
20100707	2.819	1.205	0.582	20100721	10	1.077	20100720	9	2.147	20100716	7	0.843
20100901	2.544	0.493	1.239	20100930	20	4.980	20101109	48	10.098	20100923	15	3.788
20101118	1.575	0.200	-0.223	20101130	7	-1.556	20110111	36	4.357	20101123	3	-1.295
20101201	2.269	0.947	0.173	20110128	40	4.956	20110223	56	7.348	20110222	55	8.224
20110321	1.501	-0.149	0.561	20110418	20	1.386	20110504	31	5.600	20110418	20	1.386
20111006	1.676	-0.182	2.973	20111019	9	3.466	20111101	18	4.944	20111017	7	2.512
20120713	1.621	-0.390	0.616	20120731	12	1.840	20120725	8	-0.776	20120720	5	0.366
20121119	1.650	-0.058	0.378	20121219	21	3.537	20121226	25	2.501	20121127	5	0.653
20131010	2.183	0.734	0.421	20131107	20	3.082	20131205	39	4.547	20131205	39	4.547
20141017	1.633	0.118	1.312	20141210	37	6.868	20141211	38	7.228	20141210	37	6.868
20141218	2.427	0.150	0.869	20150105	10	-1.541	20150105	10	-1.541	20150105	10	-1.541
20150203	1.759	0.038	1.199	20150306	22	1.112	20150309	23	1.890	20150209	4	0.364
20150930	1.468	-0.078	1.231	20151109	28	8.592	20151112	31	6.997	20151021	15	5.331
20161107	2.076	0.401	1.402	20170109	42	8.589	20170119	49	7.810	20170130	56	9.020
20180214	1.027	1.233	0.075	20180228	9	0.601	20180301	10	-1.078	20180220	3	0.299
20180504	1.389	0.391	0.012	20180529	16	0.442	20180531	18	0.678	20180515	7	1.824
20190604	2.065	0.819	0.709	20190725	36	6.943	20190729	38	7.239	20190717	30	7.230
				Average=	**21.667**	**3.332**	**Average=**	**29.778**	**4.271**	**Average=**	**20.389**	**3.183**

E. Experiments on Profit Estimation

Table II shows an experimental result that is performed on June 4, 2019 as the reference day whose candlesticks are shown Figure 5. Because the length of the candlestick body on the day is 1.5036%, the length of candlestick body is restricted between 1.4036% (= 1.5036 − 0.1) and 3.5036% (= 1.5036 + 2.5). The value of *SumNC* and *SumND5av* are set to −3%. Accordingly, the value of *PM5day* is 1.6% (= 1 – (–3/5)).

The other parameters are carefully adjusted to retrieve approximately 18 sample days that is suitable sample sizes to be statistically analyzed. The *x* column in Table II shows price changes of the market day *MDay*. *y* and *z* columns show price changes of the next day and the day after next. The *Date to sell*, *Holding Days*, and *Profits* columns mean the date to sell back stocks, the number of days to keep stocks holding, and simulated profits, respectively. Averages of profits are 3.332%, 4.271%, and 3.183% for *SumNC*, *SumND5av*, and *PM5day*, respectively.

Success trade ratio is calculated by dividing the number of retrieved dates that yield profits by the number of retrieved dates. Table III summarizes profit averages and success trade ratios for each parameter to trigger selling stocks. The simulated trade profit average using parameter *SumND5av* marks high profit average of 4.271%, while shows low success trade ratio of 0.833. The trade profit averages using parameters *SumNC* and *PM5day* show the opposite, i.e., rather low profit average with high success trade ratio. Product of the profit average and success trade ratio seems to suggest potential profits.

TABLE III. AVERAGES OF PROFITS AND SUCCESS TRADE RATIO

	SumNC	**SumND5av**	**PM5day**
Profit average (%)	3.332	4.271	3.183
Success trade ratio (%)	0.889	0.833	0.889
Profit average * Success trade ratio (%)	2.962	3.559	2.830

Figure 7 shows a graph with a profit in the y-axis, and a value of parameters *SumNC*, *SumND5av* and *PM5day* in the x-axis. Generally, the profits increase as the values of parameters to trigger selling stocks increase. Profits peak at the value of 4% to 6% and seem to decline over 8%.

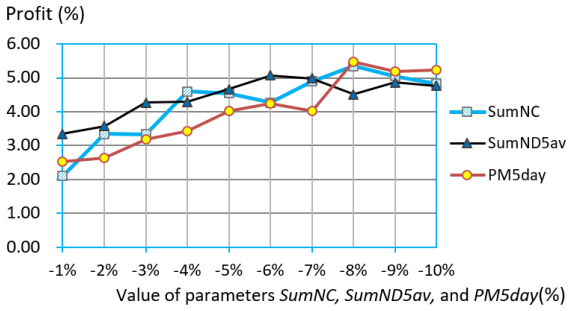

Figure 7. Graph showing exit value and profit

Figure 8 shows a graph with an average of stock holding days in the y-axis, and a value of parameters *SumNC*, *SumND5av* and *PM5day* in the x-axis. The averages in the y-axis increase approximately linearly as the values of parameters in the x-axis increase.

 ISBN: 978-1-61208-775-7

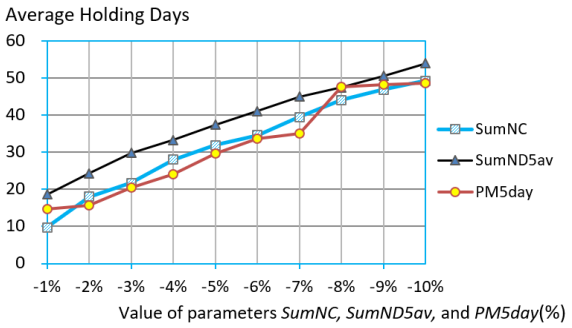

Figure 8. Graph showing exit value and average holding days

Table IV summarizes the result of the statistical analysis using Excel that is performed by specifying *Profit* as an independent variable and *Holding Days* as a dependent variable. *R Square* is 0.7647 suggests that 76.47% of *Profit* values can be explained by the variable of *Holding Days*. The last column of the table *ANOVA* shows the results of an overall *F* test. The value of *Significance F* is 0.000002074 (<0.05), which indicates that *Profit* are significantly related to *Holding Days*.

TABLE IV. RESULTS OF REGRESSION ANALYSIS

Summary Output

Regression Statistics	
Multiple R	0.8745
R Square	0.7647
Adjusted R Square	0.7500
Standard Error	1.7327
Observations	18

Analysis of Variance (ANOVA)

	df	SS	MS	F	Significance F
Regression	1	156.124	156.124	52.005	2.07407E-06
Residual	16	48.034	3.002		
Total	17	204.157			

	Coefficients	Standard Error	t Stat	P-value	Lower 95%	Upper 95%
Intercept	-1.57202906	0.9073	-1.7326	0.1024	-3.4955	0.3514
Holding Days	0.19621534	0.0272	7.2114	0.0000	0.1385	0.2539

Figure 9 shows simulated profits in a bar graph for each retrieved day using *SumNC, SumND5av, and PM5day*.

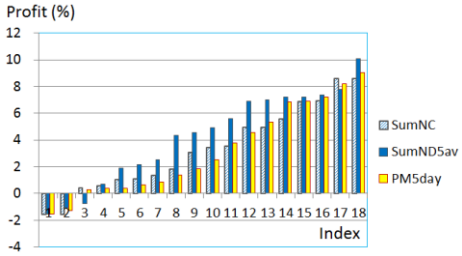

Figure 9. Calculated profits for each retrieved day (Dow Jones index)

As Figure 9 shows, the values of three parameters to trigger selling stocks produce comparable profits and/or losses. Strictly speaking, while *SumND5av* parameter yields better profits than those of the other two. *SumND5av* tends to generate larger losses while generates larger profits.

F. Experiments on Asian Markets

Experiments are performed using three Asian markets. The retrieval conditions are almost the same as those used in Dow Jones index, though in order to retrieve approximately 18 market days, values of parameters, such as *av5diff_tol*, *slope5_tol* etc., are adjusted within 0.5%.

Figure 10 shows simulated profits of Shanghai index. Five days out of 19 days result in losses. The maximum profit is estimated about 20%, which occurs on Oct. 28 2014. Other high returns of approximately 12% happen on Sept. 30, 2010 and Dec. 5, 2012. Excluding the three large profits, Shanghai market seems less profitable than Dow Jones market.

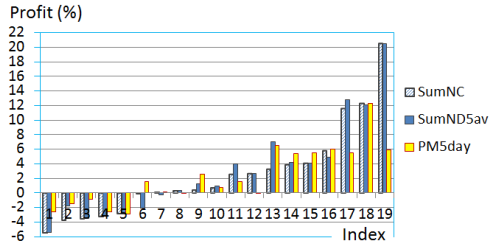

Figure 10. Calculated profits for each retrieved day (Shanghai index)

Figure 11 shows simulated profits of Hang Seng index. Four days out of 18 days result in losses. The maximum profit is estimated about 8%, which is less than that of Dow Jones index.

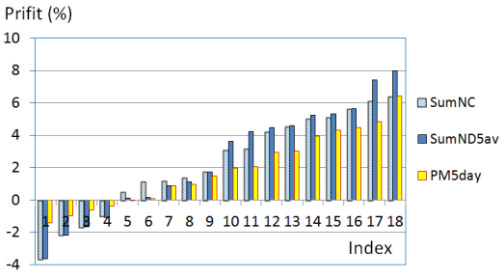

Figure 11. Calculated profits for each retrieved day (Hang Seng index)

Figure 12 shows simulated profits of Nikkei 225 index. Two days out of 18 days result in losses.

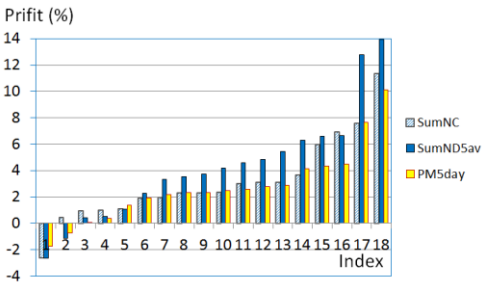

Figure 12. Calculated profits for each retrieved day (Nikkei 225 index)

Large profits of 14.0% and 12.9% occur on Sept. 11, 2017 and Nov. 15, 2012. Excluding the two profits, Nikkei 225 market seems comparable to Dow Jones market.

 ISBN: 978-1-61208-775-7

VI. Conclusion and Future Work

This paper proposes a model for retrieving similar candlestick charts with six parameters. It deals with the 5-day and 25-day moving averages to identify their trends in addition to decide whether the price occurs in price high or low zones. Since successful stock trade is significantly depends on good timing of selling stocks, three criteria are proposed and the profits that they generate are simulated using developed and emerging markets in the US and Asia.

The experiments are performed on a pattern that suggests a strong uptrend according to the prediction based on candlestick pattern, i.e., the *morning star* pattern followed by the *three white soldiers* one. Daily stock data of two US markets and three Asian markets are used in the experiments. The results show that the pattern yields significant profits in all markets. As for profits in global markets, the experimental results generally support what is stated in the paper of Dimson et al. [1].

Future work may include experiments using other candlestick patterns to measure the profitability of the proposed method. Additional studies may be conducted to compare global markets to meet demands of finding the most profitable market.

References

[1] E. Dimson, P. Marsh, and M. Staunton, "Should you invest in emerging markets?" London Business School, Apr. 2019, Available from: https://www.london.edu/think/emerging-markets

[2] N. Ahmad, R. R. Ahmed, J. Vveinhardt, and D. Streimikiene, "Empirical Analysis of Stock Returns and Volatility: Evidence from Asian Stock Markets," Technological and Economic Development of Economy, vol. 22, 2016, pp. 808–829.

[3] "Technical Analysis," Cambridge Univ. pp. 1–179, Feb. 2011, Available from: http://www.mrao.cam.ac.uk/~mph/Technical_Analysis.pdf

[4] J. M. Horton, "Stars, crows, and doji: The use of candlesticks in stock selection," Quarterly Review of Economics and Finance, vol. 49, Nov. 2007, pp. 283–294.

[5] R. B. Marshall, R. M. Young, and R. Cahan, "Are candlestick technical trading strategies profitable in the Japanese equity market?" Review of Quantitative Finance and Accounting, vol. 31, Aug. 2008, pp. 191–207.

[6] P. Tharavanij, V. Siraprapasiri, and K. Rajchamaha, "Profitability of Candlestick Charting Patterns in the Stock Exchange of Thailand," SAGE journals, Oct. 2017, pp. 1–18.

[7] G. Caginalp, and H. Laurent, "The predictive power of price patterns," Applied Mathematical Finance, vol. 5, Jun. 1998, pp. 181–206.

[8] Y.-J. Goo, D.-H. Chen, and Y.-W. Chang, "The application of Japanese candlestick trading strategies in Taiwan," Investment Management and Financial Innovations, vol. 4, Jan. 2007, pp. 49–79.

[9] C. Chootong and O. Sornil, "Trading Signal Generation Using a Combination of Chart Patterns and Indicators," International Journal of Computer Science Issues, vol. 9, Nov. 2012, pp. 202–209.

[10] C.-F. Tsai and Z.-Y. Quan, "Stock Prediction by Searching for Similarities in Candlestick Charts," Journal ACM Transactions on Management Information Systems (TMIS), vol. 5, Jul. 2014, pp. 1–21.

[11] M. Zhu, S. Atri, and E. Yegen, "Are candlestick trading strategies effective in certain stocks with distinct features?" Pacific Basin Finance Journal, vol. 37, Apr. 2016, pp. 116–127.

[12] M. Jamaloodeen, A. Heinz, and L. Pollacia, "A Statistical Analysis of the Predictive Power of Japanese Candlesticks," Journal of International & Interdisciplinary Business Research: vol. 5, 2018, pp. 62–94, Available from: https://scholars.fhsu.edu/jiibr/vol5/iss1/5

[13] Y. Udagawa, "Dynamic Programming Approach to Retrieving Similar Candlestick Charts for Short-Term Stock Price Prediction," International Journal on Advances in Software, IARIA, vol. 11, Dec. 2018, pp. 440-451.

 ISBN: 978-1-61208-775-7

Designing a Data Logistics and Model Deployment Service

Jedrzej Rybicki
Juelich Supercomputing Center (JSC)
Juelich, Germany
Email: j.rybicki@fz-juelich.de

***Abstract*—In Big Data applications, it is often required to integrate data from different sources to fuel machine learning models. In this paper, we describe a prototype implementation of the data logistics and model deployment services. Our goal was to create a one stop shop solution to support generic Data Science life cycle. It starts from formalized and repeatable data selection and processing provided by the data logistic service. The data are used for model creation in a typical machine learning fashion. The model is then put into a model repository to enable easy model management, sharing, and deployment. The functionality of the proposed prototype is positively verified with a particular use case from environmental science.**

Keywords–Data Science; Big Data; Machine Learning; Model Repository.

I. INTRODUCTION

Data Science is a way of obtaining novel insights from collected data. The process is propelled by two main forces: large amounts of data and analysis methods subsumed under the term *machine learning*. There are many ways of defining the Data Science process [1], but for the sake of argumentation, we can reduce it to three main phases: data preparation, model creation, and model deployment. Each of the phases poses some unique challenges. The modelling phase probably attracts most of the attention. This part unifies approaches from applied computer science, statistics, artificial intelligence, and many more popular scientific fields. Yet, the phase cannot be conducted efficiently without the data collection phase, and it is not very useful if the created model is not put into production. Therefore, in this paper we focus on the data collection and model deployment and propose a solution, which is sufficiently generic to accommodate different kinds of models.

The quality of the outputs of a Data Science project is mainly resulting from the quality and amounts of the input data used (rather than a sophistication of the used model). Thus, in the process of data preparation, one has to make sure to collect as much relevant data as possible. Just as in the physical world, a factory needs to be timely supplied with all the production means it requires, and the quality of the products depends on the resources used. The problem in physical world is solved by logistics. Along these lines, in this paper we propose a *data logistics service* responsible for timely delivery of the data to the models.

Collection of the data requires access to many sources. Furthermore, the data have to be cleansed to ensure their quality. Also, a higher-level processing is often required, for instance, to transform the data into a different format. In our experience, the process of data preparation takes a lot of time and effort, and yet becomes little acknowledgment because it is regarded as a mundane and less important process than modeling. The challenges posed in the phase of data collection are further reinforced by the fact that in many cases new data becomes available during the process as data collections evolve over time. This is especially important for the forecasting models, as their output might be more dependent on the most up-to-date information rather than the historical ones. The model performance depends on the *data freshness*. It is generally considered a bad practice to perform data collection and processing in a manual way [2]. Rather a formalization of the process in form of programming scripts shall be sought after. Programmatic approach helps in understanding and repeating the process of data collection and can also be crucial for efficient provenance tracking. The availability of programs and scripts for automated data collection and processing does not alone solve the problem of keeping track of data changes. Because of the aforementioned requirement of data freshness, the data collection is not a one-off act but rather a repeatable action. Thus, the programs and scripts have to be executed periodically, and monitored to detect progress, errors, and problems. In this paper, we propose an approach based on Apache Airflow [3] to implement data logistic service to gather and process data in an automatic, repeatable, and user-friendly way.

Second phase we would like to focus our paper on, is the model deployment phase. It follows the phase of model selection, tuning, and training. This is often done, at least partly, in an interactive way with tools like Jupyter Notebook [4], or Zeppelin [5]. Such tools are second to none in terms of user friendliness and quick turn over times (at least for small models). As soon as a promising model is found and its basic parameters are set, a more laborious phase of model training follows. Roughly speaking, this process sets up the model internal parameters to try to fit the collected empirical data as good as possible. Depending on the size of the data and complexity of the model the training can take substantial amounts of time. The trained model should be then put into production to accomplish the work it was intended to do. The production can be a support of an interactive web application where the model does the predictions, classifications, visualizations, etc. Given the dynamic nature of the data used in most Data Science projects, the model may require a retraining to account for the newly collected information. Sometimes also an adjustment of the parameters, or even change of the model class is required. In this paper, we show how the training of a model can be incorporated in the proposed data logistics service, and also, how the trained models can be put in MLflow [6] model repository to enable easy model sharing, review, and deployment. Our goal is to provide a one-stop shop solution to support complete Data Science life cycle.

Our high-level motivation is based on two observations. Models created in scientific endeavours should be verifiable

 ISBN: 978-1-61208-775-7

by other researchers. Such a verification can be conducted also by applying given model to a new set of data. We believe that our approach can be helpful here. Secondly, we observe increasing asymmetry between resource usage of model training and prediction. Complex models, e.g., neural networks driven by large amounts of data often require large amounts of special kinds of hardware for efficient training. Yet a prediction with such models are pretty quick even with simple hardware. Also, for these purposes, a model repository with model deployment functionality can be beneficial. It allows for large research organizations to share their (often expensive) specialized hardware and results it produces.

The rest of the paper is structured as follows. We firstly, summarize the use case that motivated our work in Section II. We then proceed with the description of the system design in Section III, where we describe both the data logistic and model deployment services and their interplay. The created solution is evaluated in Section IV. Subsequently, we shortly discuss related works in Section V, before summarizing the paper in Section VI.

II. USE CASE DESCRIPTION

In this paper, we propose a system to support typical tasks in a Data Science project. In particular, we cover the data preparation and model deployment phases. These phases occur in many standardized Data Science life cycles (even if under different names) like Cross Industry Standard Process for Data Mining [7] or Team Data Science Process [8]. To better understand how the proposed solution can facilitate efficient Data Science endeavors, let us describe a use case that motivated our implementation.

Firstly, our goal was to put the relevant data in a target database. Subsequently, the database was used for training a machine learning model, which was then put into production to conduct forecasting. Our data source was the OpenAQ Platform [9]. It collects measurement of following pollutant types PM10, PM2.5, sulfur dioxide (SO2), carbon monoxide (CO), nitrogen dioxide (NO2), ozone (O3), or black carbon (BC). OpenAQ stores raw data from measuring stations operated by government entities or international organizations across the world. The data are accessible through an API and also put in a public storage based on Amazon Simple Storage Service (S3) [10]. OpenAQ publishes data in different formats and with different time resolutions, we were interested in the most current ones, i.e., the real-time version published every 10 minutes to S3 [11].

Our target database was Tropospheric Ozone Assessment Report (TOAR), which is a relational database of global surface ozone observations emerging from a cooperation among many data centers and individual researchers worldwide. It combines data from over 10 000 measuring stations, allowing for sophisticated analysis of ozone concentrations in troposphere. Ozone is relevant for both human health and environment [12]. Access to the collected data is granted through Jülich Open Web Interface for accessing TOAR surface ozone data [13]. OpenAQ shall become one more of many sources of data integrated into the TOAR database.

Two main challenges with respect to data management were to keep them up-to-date and transform data from OpenAQ into a new TOAR format. Roughly speaking, the TOAR database is built around the notion of measurement series stored in a relational database, whereas OpenAQ collects single measurements stored in compressed NDJSON [14] format. Such discrepancies are typical in real life and have to be often addressed in the data collection phase.

The target database was used to retrieve relevant measurement series, which in turn were used to train a model for predicting air quality in a given area. The model was deployed and served as an analytic backend for a web application. We intentionally omit some details regarding the actual model and its usage, this part belongs to a different Data Science life cycle phase, which lays outside of the scope of this paper. The presented use case comes from a scientific field of environmental science, but we believe that the principles apply in other scientific fields and also outside of the academia, where Data Science approaches become more and more popular. The data flow in our use cases is schematically depicted on Figure 1.

III. SYSTEM DESIGN

In this section, we describe the design and implementation of the proposed solution. Its two main parts are data logistics service and model repository with deployment function. Although, as we pointed out, the parts support distinct phases of Data Science life cycle, there is also an overlap between them. For instance, a model can only be deployed when it passed the training phase fuelled by the delivered data.

A. Data logistics

Our solution for data logistic is based on Apache Airflow [3]. It is a platform to programmatically author, schedule, and monitor workflows. Workflows are defined as Directed Acyclic graphs (DAGs), which comprise of *Operators* and additional metadata defining, e.g., execution frequency. A unique feature of Airflow, when comparing to well-known workflow systems like Taverna [15] or Kepler [16], is that it does not use a product-specific language for defining workflows, but rather uses standard Python programming language. This allows for more flexibility in terms of task and dependencies and also lowers the entry barrier for the new users.

The way the Airflow workflows are executed differs from the aforementioned workflow systems. DAG's Operators are instantiated to become Tasks, which are then passed through a messaging queue to the Worker nodes for execution. The number of workers in the system can be changed depending on the workload. Operators abstract different kind of tasks and constitute extension points in system. There are three kind of Operators in Airflow: actions, data transfers, and sensor. Sensors wait and detect a particular event, e.g., publication of new data. Data transfer operators move data to and from particular system (like database, object store, etc.). Finally, the action operators execute particular action in remote environment, for instance `DockerOpetator`,`SparkOperator`, `BashOperator`, `SSHOperator`.

Airflow has a very unique yet powerful approach to the repeatable tasks. To enable reproducibility of the workflows their dependency on time is reduced. Each DAG has to have a `start_date` and `schedule_interval`. The `end_date` is optional and if no value is provided a date in the future is used to facilitate perpetual repeating workflows. The interval is divided into smaller parts, each of `schedule_interval` length. For each of the parts, one DAG run is created and executed. The time variable is injected into the tasks of the workflow. The tasks must be implemented in such a way that they should rely on the execution time provided by

 ISBN: 978-1-61208-775-7

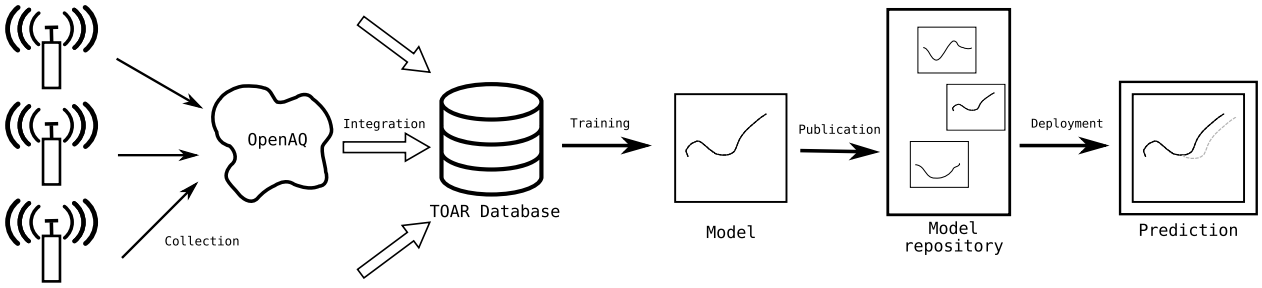

Figure 1. Data and computation flow in the modeled use case.

the workflow system (rather than other means like operating system date or time values). Thus, upon failure of a workflow it is possible to restart it at later date. Also, changes in the workflow or single tasks can be easily implemented and then rerun in an efficient way.

The workflow for the retrieval of the OpenAQ data and upload it TOAR database is composed of three main tasks. Firstly, a list of objects in S3 is created and filtered so that only objects from the injected time interval of ten minutes are considered in the next steps. Subsequently, the identified compressed NDJSON objects are downloaded and temporarily stored in a distributed file system. Lastly, the files are analyzed and uploaded to the target TOAR's Postgres database. During the last task, single measurements from OpenAQ are analyzed. If they refer to a measuring station, which is present in the target database, the measurement is added to the existing measurement series, otherwise a new series is created. Unfortunately, the stations in OpenAQ do not have a unique identifiers. Their names are given by the station operators and can even change over time. Therefore, we decided to use coordinates to identify the stations. This worked for most of the cases, stations with no coordinates were discarded.

Since new measurements are published every 10 minutes to S3 buckets, the workflow has to be rerun periodically. This part is taken care of by Airflow scheduler. To speed-up the processing, some of the tasks are implemented in a parallel fashion: NDJSON objects are split into chunks, which are processed in parallel.

B. Model creation

The availability of the data marks a starting point, at which training of a model for air quality forecast becomes possible. The process is manifold. Firstly, the relevant data are selected from the database. We are interested only in measurements regarding particular station. Secondly, a simple *RandomForestRegressor* model from Python `Scikit-learn` [17] package is trained. It is worth stressing that we use this very simple model only to show case how our solution works. In reality, the scientists doing the analysis would be using much more data and much more sophisticated models. The input data for the model is pulled from a temporary data table. The reason for this is the flexibility to use the same train code with different data. The progress of the training process is traced with help of a `MLflow` server [6]. With such a server, it is possible to store model parameters, metrics like mean square error, and model artifacts, e.g., serialized model.

We implemented the model creation as an Airflow DAG. One of the challenges of such an approach is the dependency management. Although, we use a popular Python library (`Scikit-learn`), it might not be available at the Ariflow workers executing the model training task. The `MLflow` offers help here. It is possible to create `MLflow` Projects that comprise not only of the code for the model creation but also metadata to define its dependencies. For Python, `conda` can be used, which is a well-established and mature package, dependency and environment management solution [18]. It will, upon project execution, take care of downloading and installing all the required libraries. This solution also works with other programming languages and libraries.

It is worth stressing that our approach is pretty flexible. The created Airflow DAG is capable of running different `MLflow` projects, which train the model. Such projects can be stored in GitHub [19] repositories from which they are retrieved for execution. The only constrain that we put on the projects is the convention for data retrieval. In our case, it is assumed that the data will be in placed in a temporary database table. The address of the database is injected to the `MLflow` project through environment variables. At the same time, the `MLflow` Projects can be executed outside of our data logistics service, e.g., on a local machine in a early phase of model selection.

C. Model deployment

A nice side effect of using `MLflow` [6] to store the models is an ability to instantiate such models in an easy way. For this, a single command is required:

```
mlflow models serve
  -m runs:/98ec38b6b846/model
  -p 8081
```

Each model registered with the `MLflow` has its unique run id, which can be used to instantiate it as in the command above. Models are decorated with a REST interface, which is accessible at given port (`-p 8081`). To this endpoint, a request with data in JSON format can be sent. The data will be passed over to the model for predictions and the results are sent back to the client.

IV. Evaluation

Despite this paper being a work in progress record, we decided to include some preliminary evaluation of the system performance. To this end, we used data from the OpenAQ data repository and analyzed 10 subsequent days starting from 1. January 2014. For the execution of a test system, we override Airflow configuration to reduce the task parallelism to 1 to eliminate the possibility of DAGs interleaving. Figure 2 shows the Task run times for the most important tasks in the DAG:

 ISBN: 978-1-61208-775-7

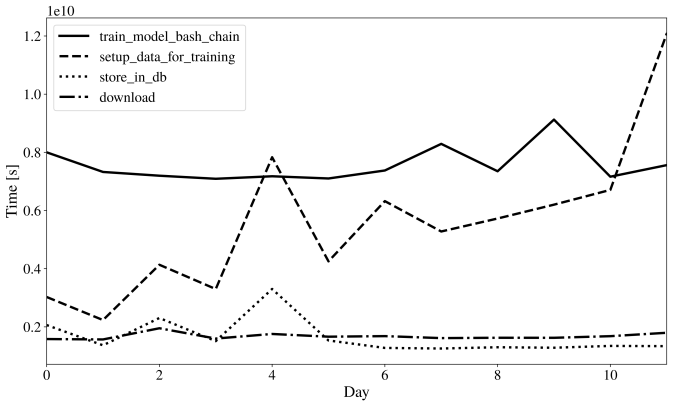

Figure 2. Duration of workflow tasks.

download of the data, data conversion and database upload, data selection for training, model training and publication. Apparently, the data download and conversion take constant amount of time, similarly to training and model upload. This is true, despite the increasing amounts of data used for the training (36 measurements on first day, comparing to 345 on the last one). The only tasks that displays some worrisome scalability characteristics, is the data preparation step in which all measurements from one selected measuring station are retrieved from a database and made available for training. Perhaps the negative scalability trend can be overcame by some database optimization and shall be a subject of following works.

V. Related Work

Our work is partly motivated by the publication of Chen et al [20] who proposed Data-as-a-Service (DaaS) and Analytics-as-a-Service (AaaS). Our data logistic service could be understood as a type of DaaS, and some functionalities of the model repository can be used to offer Analytics services.

We already pointed out the differences between used solution for data logistics and typical workflow systems like Taverna [15] or Kepler [16] (see Section III). Airflow addresses different kind of use cases, focuses mostly on efficient data movement and integration. We think, that the proposed solution is orthogonal to the classical workflow systems, the later ones can be executed by Airflow, e.g., to facilitate model training where computation performance is of the primary interest.

The problem of model repository and efficient model deployment seems to be gaining attention lately. There is a patent describing an idea of model repository [21]. It provides, however, no implementation details. FBLearner Flow by Facebook [22], Google TensorFlow Extended [23], or Kubeflow [24] are infrastructure and framework specific solutions for model deployment and management. In our work, we were striving for a generic solution and also wanted to promote the idea of model sharing in academia. The Open Science movement was successful in promoting the idea of data repository. A truly open science requires publishing and sharing of the created models. An interesting work by Behrouz [25] focuses on continuous model deployment. At first glance the approach is complementary to the solution presented herein. The author provided a means for efficient model retraining, whereas we cover the remaining phases of collecting data, changing and redeploying of the model. We intend to verify if the proposed solution can alleviate the problem of limited scalability of data selection tasks observed in our evaluation.

VI. Conclusion and Future Work

In this paper, we proposed a prototypical implementation of the data logistics and model deployment services. The services were used to implement a real use case from environmental science. We described and explained our design decisions. The use case was implemented successfully and put into production.

In our future work, we plan to address some of the shortcomings of our current solution. More improvement is required in terms of efficient sharing of Airflow DAGs. This will help in implementing other use cases. Also, more sophisticated deployments (e.g., with monitoring and dynamic resource management) are planned. To this end, migration to Docker-based model execution might be a good idea.

References

[1] J. Rybicki, “Best practices in structuring data science projects,” in Proceedings of the International Conference on Information Systems Architecture and Technology. Springer, 2018, pp. 348–357, ISBN: 978-3-319-99992-0. [Online]. Available: https://doi.org/10.1007/978-3-319-99993-7_31

[2] G. Wilson et al., “Good enough practices in scientific computing,” PLOS Computational Biology, vol. 13, no. 6, Jun. 2017, pp. 1–20. [Online]. Available: https://doi.org/10.1371/journal.pcbi.1005510

[3] Apache Airflow. [Online]. Available: https://airflow.apache.org/ [retrieved: 2020. 01 .01]

[4] Project Jupyter. [Online]. Available: https://jupyter.org/ [retrieved: 2020. 01 .01]

[5] Apache Zeppelin. [Online]. Available: https://zeppelin.apache.org/ [retrieved: 2020. 01 .01]

[6] MLflow. [Online]. Available: https://mlflow.org [retrieved: 2020. 01 .01]

[7] P. Chapman et al. CRISP-DM 1.0: Step-by-step data mining guide. [Online]. Available: ftp://ftp.software.ibm.com/software/analytics/spss/support/Modeler/Documentation/14/UserManual/CRISP-DM.pdf [retrieved: 2020. 01 .01]

[8] TDSP: Team data science process. [Online]. Available: https://docs.microsoft.com/en-us/azure/machine-learning/team-data-science-process/overview [retrieved: 2020. 01 .01]

[9] OpenAQ platform (openaq.org). [Online]. Available: http://openaq.org/ [retrieved: 2020. 01 .01]

[10] D. Robinson, Amazon Web Services Made Simple: Learn How Amazon EC2, S3, SimpleDB and SQS Web Services Enables You to Reach Business Goals Faster. London, UK, UK: Emereo Pty Ltd, 2008, ISBN: 978-1-92157-306-4.

[11] OpenAQ daily fetches at S3. [Online]. Available: https://openaq-fetches.s3.amazonaws.com/index.html [retrieved: 2020. 01 .01]

[12] M. G. Schultz et al., “Tropospheric ozone assessment report: Database and metrics data of global surface ozone observations,” Elementa: Science of the Anthropocene, vol. 5, 2017, pp. 58–68. [Online]. Available: https://doi.org/10.1525/elementa.244

[13] Jülich Open Web Interface for accessing TOAR surface ozone data. [Online]. Available: https://join.fz-juelich.de/ [retrieved: 2020. 01 .01]

[14] NDJSON - Newline delimited JSON. Specification. [Online]. Available: https://github.com/ndjson/ndjson-spec [retrieved: 2020. 01 .01]

[15] T. Oinn et al., “Taverna: a tool for the composition and enactment of bioinformatics workflows,” Bioinformatics, vol. 20, no. 17, 06 2004, pp. 3045–3054, ISSN: 1367-4803. [Online]. Available: https://doi.org/10.1093/bioinformatics/bth361

[16] B. Ludäscher et al., “Scientific workflow management and the Kepler system,” Concurrency and Computation: Practice and Experience, vol. 18, no. 10, 2006, pp. 1039–1065. [Online]. Available: https://doi.org/10.1002/cpe.994

 ISBN: 978-1-61208-775-7

[17] F. Pedregosa et al., “Scikit-learn: Machine learning in Python,” Journal of Machine Learning Research, vol. 12, 2011, pp. 2825–2830, ISSN: 1533-7928.

[18] Conda. [Online]. Available: https://conda.io/ [retrieved: 2020. 01 .01]

[19] GitHub. [Online]. Available: https://github.com/ [retrieved: 2020. 01 .01]

[20] Y. Chen, J. Kreulen, M. Campbell, and C. Abrams, “Analytics ecosystem transformation: A force for business model innovation,” in Proceedings of the IEEE Annual SRII Global Conference, Mar. 2011, pp. 11–20, ISSN: 2166-0778.

[21] C. R. Chu and S. C. Tideman, “Model repository,” Patent US Patent 6,920,458, 2005.

[22] FBLearner Flow. [Online]. Available: https://engineering.fb.com/core-data/introducing-fblearner-flow-facebook-s-ai-backbone/ [retrieved: 2020. 01 .01]

[23] TensorFlow Extended (TFX). [Online]. Available: https://www.tensorflow.org/tfx/ [retrieved: 2020. 01 .01]

[24] Kubeflow. [Online]. Available: https://www.kubeflow.org/docs/about/kubeflow/ [retrieved: 2020. 01 .01]

[25] B. Derakhshan, “Continuous deployment of machine learning pipelines,” in Proceedings of 22nd International Conference on Extending Database Technology (CEDT’ 19), Mar. 2019, pp. 397–408.

 ISBN: 978-1-61208-775-7

DFSCC: A Distributed Framework for Secure Computation in the Cloud

Mamadou H. Diallo, Christopher T. Graves, Michael August, Verinder Rana, and Kevin Groarke

Naval Information Warfare Center Pacific, San Diego, CA USA
U.S. Department of Defense
Email: {mamadou.h.diallo, christopher.t.graves, michael.august, verinder.rana, kevin.groarke}@navy.mil

***Abstract*—Currently, various advanced data analytic tools based on machine learning and data mining techniques are available for performing data analysis in the cloud. However, these tools are not very secure since the data they operate on must be in plaintext, thereby leaving the data vulnerable to both insider and outsider attacks. In this paper, we take a different approach and propose the Distributed Framework for Secure Computation in the Cloud (DFSCC), a flexible framework for building secure, distributed computation and sharing systems. The framework takes advantage of Homomorphic Encryption (HE) techniques to enable data analytics to be performed directly on the encrypted data stored within the nodes of the distributed system. An advantage of distributing data analytics into the nodes of the framework is enhanced performance of HE-based computation. In addition, the framework incorporates a cryptographic key management infrastructure to enable secure data sharing. To evaluate the framework, we extended it to implement a system that analyzes link quality between software defined radios using a machine learning algorithm. Experiments performed on the system show performance improvement of the system as the number of nodes in the cluster is increased.**

Keywords–Homomorphic Encryption, Secure Computing, Privacy, Machine Learning, Distributed Systems

I. INTRODUCTION

Performing data analytics in the cloud is becoming increasingly significant for organizations of all types and sizes. The cloud provides the scalable infrastructure and resources needed to efficiently run the analytic tools. Organizations are taking advantage of these analytic tools to gain powerful insights out of the ever-growing pools of organizational data. These analytics are in general based on techniques such as machine learning, data mining, and statistical analysis [1]–[3]. These cloud based data analytic tools are being developed for various application domains [4]–[7]. However, there is also a growing concern about data security and privacy in cloud-based systems and applications that provide analytic tools [8]–[11]. In particular, cybersecurity attackers are becoming more sophisticated, and attacks on data in large organizations are occurring more frequently [12].

The main issue is how the data is manipulated by analytic tools in the cloud, which is inherited from the shortcomings of current cryptographic techniques for securing data. The recommended randomized encryption schemes, such as the Advanced Encryption Standard (AES) and Blowfish, provide strong protection of data in transit and at rest, but do not protect data in processing. This means that data needs to be decrypted in memory before processing of the data can take place, which leaves the data vulnerable to attacks from both internal and external attackers.

To address this shortcoming of existing standard cryptographic schemes, Homomorphic Encryption (HE) has been proposed [13]. HE schemes have revolutionized data security, as they enable computation to be performed directly on the encrypted data without needing the private decryption keys. Given ciphertexts as input, HE allows computation to be performed directly on the ciphertexts to generate encrypted results. When these encrypted results are then decrypted, they yield the correct plaintext answer for the computation as if it were performed entirely in plaintext. However, HE, while significantly improving data security in untrusted environments, comes with significant computation and storage overheads [14]. In general, the computational complexity of HE is orders of magnitude higher than that of standard operations on plaintext. A given ciphertext encoding is also much larger than its corresponding plaintext.

In this paper, we introduce the Distributed Framework for Secure Computation in the Cloud (DFSCC), a distributed framework that enables the development of secure computation and sharing systems using HE schemes. HE schemes provide data security not only in transit and at rest, but most importantly, during processing. The framework is modularized and extensible to enable the incorporation of different types of HE schemes. The framework also provides mechanisms for incorporating data analytic tools that use HE schemes into the nodes of the distributed framework. This enables data analytic tools to operate directly on the encrypted data. To enable data sharing, the framework includes a cryptographic key management infrastructure based on the approach introduced in [15]. Using this approach, an organization can analyze data in the cloud and share the results with other organizations. Distributing analytic tool execution into the nodes of the framework speeds up the expensive operations of the HE schemes to improve the overall performance of the tools. The framework enables tool developers using various machine learning and data mining algorithms to use the framework to build analytic tools, in addition to enabling system developers to leverage these analytic tools within their applications. The secure systems developed based on the framework can then be made available to end-users to analyze their data securely and privately in the cloud. Having access to different analytics will enable end-users to trade off between the quality of the results of the data analysis and the time it takes to perform the analysis. Furthermore, end-users will have the ability to share their data with other parties securely and privately.

The paper is organized as follows. In Section II, we describe the challenges in processing data and our proposed solution. In Section III we describe our overall approach. In

 ISBN: 978-1-61208-775-7

Section IV we outline the data flow through the system. In Section V we discuss our implementation of the framework and sample application. In Section VI we present the results of experiments performed on the system. In Section VII we contrast our paper with related works. We end the paper with a conclusion and future work in Section VIII.

II. BACKGROUND

In this section, we take a look at how organizations make use of data analytics in the cloud and give an overview of Homomorphic Encryption, which can be used to address data security in the cloud.

A. Data Analysis in the Cloud

Machine learning, data mining, and statistical modeling and analysis techniques are steadily making their way into enterprise applications in areas, such as customer support, fraud detection, and business intelligence [4]. The major cloud service providers are responding to this need for tools that provide data analysis and business intelligence capabilities within the cloud by adding these features to their cloud services [16]. Thus, the trend of organizations outsourcing their Information Technology operations to the cloud, combined with the trend of cloud service providers adding more intelligence to their cloud services, indicates that increasingly organizations will make use of the cloud to analyze their large and potentially sensitive data sets. However, the cloud is vulnerable to cyber attacks from both internal and external attackers. To address these vulnerabilities of the cloud, we use HE to ensure the confidentiality of the data that are collected, stored, and processed in the cloud.

B. Homomorphic Encryption

Craig Gentry [13] introduced the first working *Fully Homomorphic Encryption* (FHE) scheme in his 2009 PhD dissertation by taking a *Somewhat Homomorphic Encryption* (SHE) scheme and "squashing" the decryption circuit to reduce the noise in a process called "bootstrapping". However, this process was impractical due to the required computation time. A more practical approach explored within the FHE research community has been the *Learning With Errors* (LWE) problem and its variants, particularly the *Ring Learning With Errors* (RLWE) problem. Below, we describe these two approaches.

1) Gentry's FHE Implementation with Ideal Lattices: A *Point Lattice*, or *Lattice* $\mathcal{L}$, is the set of all integer linear combinations of a set of linearly independent vectors $\mathcal{B} \subseteq \mathbb{R}^m$. For constants $c_i \in \mathbb{Z}$ and $\mathbf{b}_i \in \mathbb{R}^m$,

$$\mathcal{L} = \sum_{i=1}^{n} c_i \mathbf{b}_i.$$

Gentry's FHE scheme encrypts a plaintext by placing it in the *fundamental region* of a lattice with "noise" generated by several classic, hard lattice problems. After a number of additions and multiplications, the ciphertext noise risks becoming so great that the ciphertext is moved outside of the fundamental region. Therefore, Gentry "squashes" the decryption circuit to give bootstrapability. An encryption scheme is *bootstrappable* if it can homomorphically evaluate its own decryption circuit. The process of bootstrapping involves providing the secret key that has been re-encrypted with a new key. In order to keep the noise at a manageable level, the bootstrapping process is done before the noise reaches the threshold where decryption is no longer possible. Because of this limitation, bootstrapping may be every other operation. In later implementations of Gentry's FHE scheme, the performance of the bootstrapping process has been optimized.

2) FHE Based on Ring Learning With Errors: To describe RLWE, let $n = 2^k$ and choose a prime modulus q such that $q \equiv 1 \mod 2n$. Let the ring $R_q = \mathbb{Z}_q[x]/\langle x^n + 1\rangle$, represent the set of all the polynomials over the finite field $\mathbb{Z}_q$ for which $x^n \equiv -1$. Given samples of the form $(\mathbf{a}, \mathbf{b} = \mathbf{a} \times \mathbf{s} + \mathbf{e}) \in R_q \times R_q$ where $\mathbf{s} \in R_q$ is a fixed secret vector, an element $\mathbf{a} \in R_q$ is chosen uniformly, and $\mathbf{e}$ is chosen randomly from an error distribution in R_q. Given this definition of the RLWE problem, finding s is infeasible.

Using the RLWE problem described above, a message $m \in R_q$ can be encrypted by using the b element above as a one-time pad encryption scheme [17]. The ciphertext can be represented by $\mathbf{c} = \mathbf{b} + \mathbf{m}$, where $\mathbf{c} \in R_q$. FHE schemes based on the infeasible RLWE Problem have been shown to be cryptographically secure given an appropriate security level.

One major effort in FHE using this approach is Microsoft's Simple Encrypted Arithmetic Library (SEAL), which utilizes the BFV FHE scheme based on RLWE [18]. The BFV scheme allows for modular arithmetic to be performed on encrypted integers. The SEAL library also implements the CKKS FHE scheme, which supports approximate arithmetic over complex and real numbers [19]. Another major FHE library which implements these schemes is the PALISADE library [20].

III. FRAMEWORK

In this section, we describe the architecture of the proposed framework, how HE schemes and data processing algorithms are integrated into the architecture, and a mechanism for data sharing.

A. Architecture

The DFSCC framework is designed using a hybrid client-server/distributed model, where clients send requests to the server, and the server sends the client's requests to the distributed system for processing. The high-level design of the framework is presented in Figure 1. The architecture is composed of two main components: *Trusted Client* and *Untrusted Cloud Environment*. We adopt the *honest-but-curious* adversarial model. We assume that the client-side is trusted while the cloud environment is untrusted. Therefore, all the private keys for decrypting the data remain with clients, and only public keys are sent to the cloud.

The *Trusted Client* comprises three main sub components, *Client Manager*, *HE Manager*, and *Configurations Manager*. The *Client Manager* coordinates the activities of the client and manages the interactions with the server. The *HE Manager* provides support for HE operations including generation and storage of public and private keys, encryption and decryption of data, and keys revocation. The *Configurations Manager* keeps track of the cloud resources for the clients, which change dynamically as the system is being used. Note that system developers will need to extend the framework to build concrete systems for specific application domains. In addition to the above core components, system developers need to implement a user interface for end-users to interact with the system.

 ISBN: 978-1-61208-775-7

The *Untrusted Cloud Environment* is composed of an *Untrusted Server* and an *Untrusted Distributed System*. All the data sets sent by the clients to the *Untrusted Cloud Environment* will remain encrypted at all times. The sub-components of the *Untrusted Server* include a *Service Engine* for coordinating all the activities related to distributing data and operations into the *Untrusted Distributed System*; an *HE Manager* for managing HE libraries stored in the *HE Libraries* storage; an *Analytics Manager* for managing the analytic algorithms persisted in the *Libraries* storage; a *Sharing Manager* for sharing encrypted data between the clients; and a *Configurations* storage for storing various cloud configurations and metadata. The *Service Engine* communicates with the *Untrusted Distributed System* to coordinate its activities, including sending workloads and partitioning the nodes within the cluster.

The *Untrusted Distributed System* provides the infrastructure for distributing analytics algorithms. The inputs to the *Untrusted Distributed System* include the set of data to be processed and the software program to be executed on the nodes of the distributed system that will process the data. At the core of the distributed system is a *Distribution Manager*, which provides the mechanisms for generating the clusters of distributed nodes. The nodes are generated by the *Distribution Manager* on demand based on the configurations provided by developers. In addition, the *Untrusted Distributed System* provides an interface to enable interaction with other distributed systems.

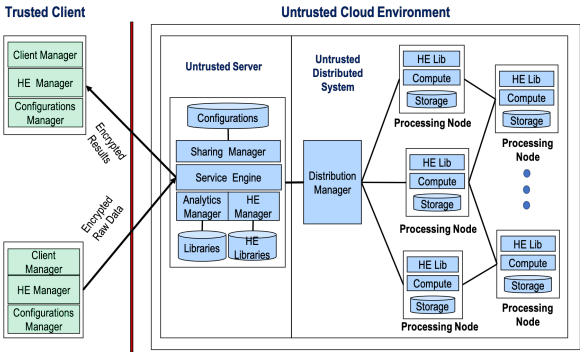

Figure 1. Distributed Framework Architecture

B. HE Scheme Integration

At the core of DFSCC are the mechanisms for incorporating HE schemes with distributed data processing algorithms. DFSCC also provides a key management infrastructure to enable sharing of data processed by the distributed system. Additionally, by abstracting out the core functionality that is commonly found in HE schemes, DFSCC is designed to facilitate the incorporation of different HE libraries. These common operations include key generation, encryption, decryption, and parameter selection. These operations are abstracted out into an interface that can then be used to integrate a given HE library.

C. Data Processing Integration

Machine learning algorithms are being used in the cloud to analyze data stored in the cloud and to enhance the cloud's capabilities. DFSCC provides an extensible interface to enable developers to extend or customize DFSCC to add new machine learning and data mining algorithms into the framework. Considering the complexity of using existing HE libraries, the first machine learning algorithm we considered for the DFSCC framework is the linear Support Vector Machine (SVM). In the future, we plan on adding more machine learning algorithms into the framework.

1) Support Vector Machines: SVMs are supervised learning models that can be used to analyze data based on classification and regression analysis. The SVM serves as a non-probabilistic binary linear classifier.

Consider a set S of sample data elements, and two subsets S_A and S_B of S, where $S_A \cup S_B = S$, and each element of S ($S_1 \in S$) is annotated as belonging to S_A or S_B. The SVM training algorithm generates a mathematical model that can be used to categorize new elements of S as belonging to S_A or S_B.

First, we are given a labeled training dataset of n points of the form $(\vec{x}_1, y_1), \ldots, (\vec{x}_n, y_n)$. This training dataset contains both the inputs and the desired outputs. Given the training dataset, we then compute the SVM model to be used for classification. This model then separates the elements of S into two classes, S_A and S_B, based on the classifier that was generated from the training data. The internal operations of the linear SVM include the dot product of vectors, addition, and subtraction. To demonstrate the utility of the DFSCC framework, we implemented an SVM classifier on top of our distributed framework using the PALISADE library.

D. Key Management Infrastructure

We use a key management system based on Public Key Infrastructure (PKI) to provide clients with mechanisms to generate, store, distribute, and revoke public/private keys in the distributed system. Overall, the key management system is based on the simple approach proposed in [15], which doesn't require a central authority for managing the keys. The approach is to exchange private keys using an email infrastructure, where each client is equipped with a built-in email server.

The protocol for exchanging public keys is as follows.

1) Exchanging Public Keys: The first time two clients, c_i and c_j, interact in the distributed system, they exchange their public keys as follows. The client c_i sends a message to c_j containing the tuple (Id_{c_i}, pk_{c_i}) and the client c_j replies with a message containing the tuple (Id_{c_j}, pk_{c_j}).

2) Data Partitioning: To facilitate data sharing, each client needs to partition their data based on sharing policies. Each partition will be encrypted using a different public/secret key pair. For instance, let's assume the user data d is divided into a set of partitions $\{d_1, d_2, \ldots, d_n\}$. Then, for each d_i, a public/secret key pair, (pk_i, sk_i) will be generated to encrypt d_i. This will give the client a flexible approach for sharing their data in the cloud at a fine-grained level.

3) Sharing Data: When a sender wants to share a piece of data d_i with a receiver in the distributed system, the sender needs to provide the receiver with the secret key sk_i corresponding to pk_i used to encrypt the data d_i in order to decrypt it. To protect the secret key, the sender encrypts it using the receiver's sharing public key. The sender replies with the following message containing the tuple $(Id_{receiver}, Enc(sk_{sender}, pk_{receiver}))$, where

 ISBN: 978-1-61208-775-7

$Enc(sk_{sender}, pk_{receiver})$ means that the sk_{sender} is encrypted using the $pk_{receiver}$. This will guarantee that only the intended receiver can decrypt the message containing the secret key. Note that, in this approach of data sharing, we assume that the distributed system includes an access control enforcement mechanism to give access to data based on sharing policies defined by the users. The description of the access control mechanism is beyond the scope of this work.

IV. DFSCC Operational Flows

The architecture of the DFSCC framework comprises a number of components that interact to support the functionalities of the framework from the perspective of both developers and end-users. It abstracts out the complexity related to building a web-based client-server application, building a cloud-based distributed system, and connecting the two. In the following sections, we describe the operational flows of the framework, focusing particularly on how developers can extend the core components of the framework and instantiate it to build concrete systems, and then discuss how end-users can use those concrete systems.

A. Extending the Framework

For developers extending the framework, there are two main features: adding a new HE library, and adding a new data processing algorithm based on machine learning or data mining techniques. At the design level, the framework employs a modular design to isolate the HE libraries and data processing algorithms. At the implementation level, the framework uses containers to enable each HE library and each data processing algorithm to be self-contained. To add an HE library, the developer needs to deploy the HE library in a container and expose an API to enable the HE manager to make use of it. Similarly, a new data processing algorithm needs to be implemented and made available to the analytics manager, which will distribute it to the nodes at runtime. In addition to the SVM implementation, other data processing algorithms will be included in the framework to serve as a guideline for developers to incorporate their own algorithms into the framework.

B. Instantiating the Framework to Build A Concrete System

The framework provides building blocks that can be used to build concrete distributed systems where analytic tools can be run in the encrypted domain. The application domain will determine the specific analytic tools to be applied using one of the available HE-enabled machine learning or data mining algorithms. For instance, in the application we built to evaluate the framework, SVM was determined to be suitable to implement a tool to analyze radio data to optimize the link quality between Software Defined Radios (SDRs). Analyzing radio link quality requires classifying the data into two classes, high quality and low quality. During the analysis, each data point falls in one of those two classes. The application domain will also dictate the type of data that needs to be encoded appropriately to ensure compatibility with the data format of the underlying HE library. Recall that the current HE libraries support only low level operations, such as addition or multiplication of numbers. It is the task of the developer to figure out how the specific data types of the application domain can be transformed in such a way that the basic operations of HE can be applied on the data.

C. Using the Concrete System

Once the system is completed, then it can be made available to end users. There are two main workflows of the system for the end user: 1) analyzing data using an analytic tool, and 2) sharing data with other users. At a high-level, the following operational workflow depicts the process for analyzing data in the distributed system.

- The User opens the Client web-based GUI.
- From the Client GUI, the user uploads the raw data to the Client local storage.
- The User selects the analytic tool to be used to process the raw data.
- The User requests the data to be encrypted.
- The Client Engine selects the appropriate HE library, and uses it to encrypt the data.
- The Client Engine sends the encrypted data along with the user parameters to the Untrusted Server.
- The Untrusted Server selects the number of nodes to use in the distributed system.
- The Untrusted Server partitions the data according to the parameters selected by the user and pushes it to the nodes.
- The Untrusted Server notifies the User after the data has been distributed.
- The User requests data to be processed and forwarded to the Untrusted Server.
- The Untrusted Server delegates the workload to the Distribution Manager.
- The Distribution Manager initiates the data processing throughout the Untrusted Distributed System.
- After the execution is completed, the Untrusted Server gathers the results from the Distribution Manager, and sends them to the User.
- The Client Engine decrypts the results and displays them on the GUI.

The following operational workflow summarizes the process for sharing data in the distributed system. The Sharing Manager on the Untrusted Server is responsible for sharing encrypted data and encrypted secret keys between parties sharing data with each other. If the recipient doesn't already have the secret key to decrypt the data, then the Sharing Manager will request the secret key from the sender, and the sender will encrypt the secret key using the recipient's sharing public key and send it to the Sharing Manager, which serves as the proxy between sender and receiver. We assume that the user possesses a public/secret key pair to be used by the underlying sharing protocol. We assume that each party has the sharing public key of the receiver. We also assume the data to be shared is stored with the Distribution Manager component.

- From the Client GUI, the sender selects the set of data to be shared, the recipients and their sharing public keys.
- The User sends the request to share the data to the Untrusted Server.
- The Sharing Manager on the Untrusted Server passes a message to the recipient containing a reference to the stored encrypted data.

 ISBN: 978-1-61208-775-7

- The Sharing Manager notifies the recipients about the availability of the data.
- The Recipients retrieve the shared data and use their secret keys to decrypt the data.

V. IMPLEMENTATION

We implemented the overall DFSCC framework and the Software Defined Radio link quality analysis application to evaluate the framework. We leveraged a number of open-source projects for the implementation including the Django web framework, PALISADE HE library [20], Apache Hadoop, Apache Spark, and Xen hypervisor.

A. Framework

The implementation is broken down into three main subsystems: client, server, and distributed system. We use the Django web framework to develop a web-based system to connect the client and server subsystems and to provide web service capability to DFSCC.

As mentioned previously, we selected the PALISADE HE library as the first library to be integrated with the DFSCC framework. PALISADE is implemented using C++ and provides a simple interface to access its basic functionality. The integration of this HE library into our framework required building a C++ wrapper to interact with the Django web server written in Python as well as the Spark interfaces used for the distribution. We used Apache Spark as the basis to implement the distributed system. Spark is highly modularized, which simplifies its integration with other systems. Spark is an ideal distribution framework for DFSCC, as it enables the distribution of data as well as programs for execution on the cloud nodes. REST APIs allow developers to extend DFSCC to build concrete applications, such as adding new HE libraries and data processing algorithms.

We used the Xen hypervisor to deploy a local instance of a cloud infrastructure as a service (IaaS). This local cloud serves as the testbed to generate virtual machines for the distributed system. We used this local cloud instance to deploy and test our distributed framework.

B. Framework Use Case: SDR Link Quality Analysis

For the test application, we implemented two Graphical User Interfaces (GUI), one for the administrator, and another for the user. Through the admin GUI, among other functionalities, the admin can create nodes (VMs) and list the resources available on the distributed system. Likewise, through the user GUI, users can upload data, encrypt and decrypt data, and send encrypted data to the cloud for processing. Once the data has been uploaded, there is a library of standard machine learning algorithms that the user can select to run on the uploaded data. Once selected, the distributed machine learning algorithm with the HE implementation will be run on the distributed system that will then return the answer in encrypted form to be decrypted when needed. We used this process to analyze the quality of Software Defined Radio signals to determine the best way to tune the radios. This was done using a simple SVM on the data to classify good versus bad link quality of the radios. Note that, in this paper, we focus only on analyzing the performance and overhead of the underlying HE operations.

VI. EXPERIMENTS

As part of the experimental setup, we deployed two Software Defined Radios, a sender and a receiver, and established a connection between the two. Then, we initiated a video stream from the sender to the receiver and extracted the data packets in the stream using the network packet capture feature of Snort. The most relevant features in this dataset are the bit error ratio, signal level, noise level, distortion level, and signal-plus-noise-plus-distortion to noise-plus-distortion ratio. Due to the limited number of features in this dataset, we generated synthetic radio data to analyze the performance of the system with a larger number of features. In the synthetic dataset, the number of features ranges from 8 to 256. We also varied the number of nodes in the distributed system from 4 to 64.

Based on the above setup, we performed a number of experiments to analyze the performance of the DFSCC framework in running the SVM based analytic tool against the encrypted dataset. Specifically, we looked at the overhead incurred by the framework due to the expensive HE operations. During the experiment, the data was grouped into varying numbers of features as follows: 8, 16, 32, 64, 128, 256. The distributed system was configured with varying numbers of nodes as follows: 4, 8, 16, 32, 64. Note that both of these scales are logarithmic as represented by the y-axis in each of the figures below. Then, we ran the analytic tool with each feature size on each node configuration.

A. Features Comparison

In this experiment, we ran all the feature sizes on 1-node, 8-node, and 16-node configurations, and recorded the running time of the computations. In both, Figure 2 and Figure 3, each of the points represents a single run of the SVM algorithm for a feature size specified on the y-axis. The x-axis represents the running time of the algorithm, where the blue circle represents a computation distributed across 8 or 16 nodes, and the red triangle represents running on a single node.

As can be seen in Figure 2, with low numbers of features, there isn't a significant difference between the two setups. However, as the number of features gets larger, the speed improvement becomes clear. Note that the outlier seen in each of the runs was an initialization of the distributed system that was exacerbated in the distributed setting but can also be seen in the single node setting. This outlier is only on the first run of the algorithm so will be less significant over many runs of the algorithm.

B. Nodes Comparison

In this experiment, we look at the comparison of running the tool with 64 and 256 features while varying the number of nodes as described earlier. We can see in Figure 4 and Figure 5 that by increasing the number of nodes there is a performance improvement. Given this, we can determine the optimal number of nodes needed for a dataset, for a given workload on the system. There is still a significant performance improvement resulting from distribution across multiple nodes, but care should be taken to balance the workload evenly throughout the distributed system in order to minimize the amount of downtime waiting for dependent computations to complete before the results of the overall computation can be delivered.

 ISBN: 978-1-61208-775-7

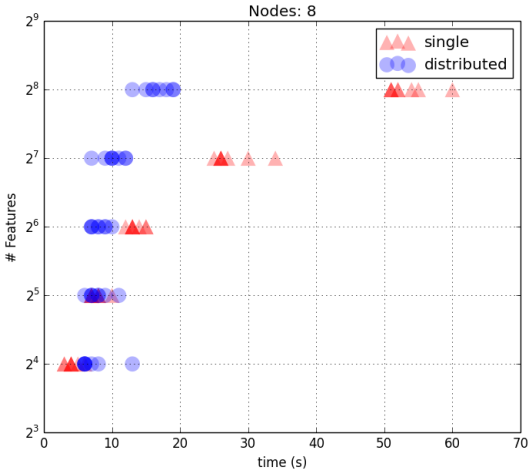

Figure 2. Overhead for 8 Nodes Setup

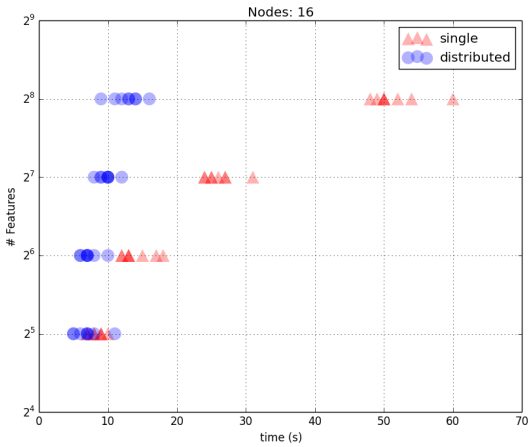

Figure 3. Overhead for 16 Nodes Setup

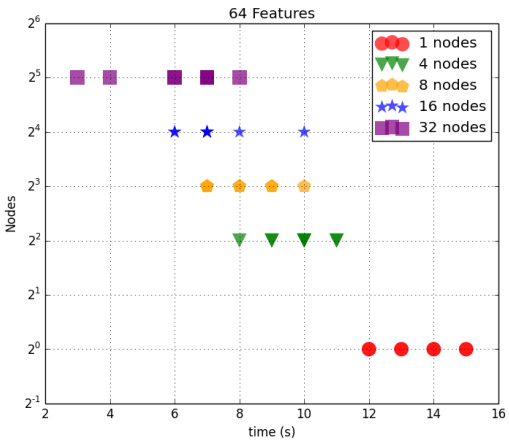

Figure 4. Overhead of 64 Features Comparison

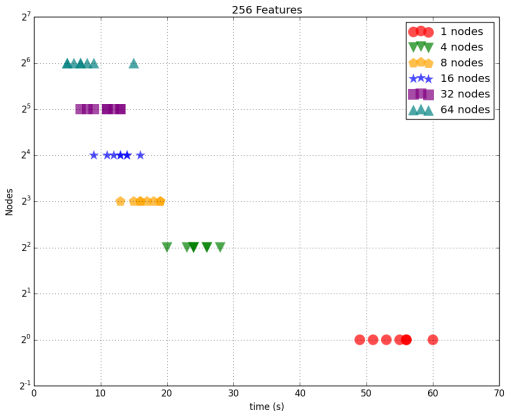

Figure 5. Overhead of 256 Features Comparison

VII. Related Work

Using HE to enable machine learning algorithms, including deep learning, to process data securely has gained attention in the research community in recent years. Many of the proposed approaches focus on using a given HE scheme to implement a specific machine learning algorithm. In [21], the authors show that it is possible to use a SHE scheme to implement a linear SVM to classify images for facial recognition. They extended Gentry's SHE scheme to work with low-degree polynomial functions, which are not limited by Hamming distance or linear projection. In [22], the authors went further by proposing an approach for implementing a non-linear SVM for classifying images in general using a SHE scheme. CryptoNets [23] uses the Microsoft SEAL HE library to implement deep learning algorithms. HE parallelization is limited to SIMD operations provided by the HE scheme. Faster CryptoNets [24] improves the performance of CryptoNets by leveraging the sparse representations throughout the neural network to optimize the HE operations and improve their performance. MSCryptoNet [25], based on multi-scheme FHE, protects the evaluation of the classifier, where the inputs can be encrypted with different encryption schemes and different keys. Unlike the above approaches, we are proposing a general framework for secure machine learning in the cloud. Furthermore, we are using distributed processing to improve the performance of machine learning computations and HE based computations.

HE schemes have also been considered as a means for securing statistical computations. In [26], the authors demonstrate the feasibility of using HE in approximating conventional statistical regression methods. This approach takes advantage of the fact that estimation and prediction can both be performed in the encrypted domain; bootstrapping can be avoided even for moderately large problems; and scales linearly with the number of predictors. In [27], HE is used to develop a secure system that protects both the training and prediction data in logistic regression. Despite the non-linearity of both the training and prediction in logistic regression, this paper showed that it is feasible to use HE since only the addition operation is needed, which significantly improves performance compared to FHE. Our approach differs in that it provides a framework to enable developers to use a variety of analytic tools which can be based on statistical analysis or other analytic techniques.

Privacy-preserving data splitting is another approach proposed to preserve data privacy in the cloud. In this approach,

 ISBN: 978-1-61208-775-7

sensitive data is split in such a way that any partition by itself is not sensitive, and is stored separately. However, the techniques proposed are not very secure as they either don't support encryption or they support only limited operations to take place in the encrypted domain [28], [29]. Furthermore, these techniques are focusing more on preserving privacy of the data at rest rather than in processing. Other proposed techniques for securing machine learning algorithms are based on MPC [30]. Fundamentally, MPC requires interactive communications among the different nodes to perform the computations, whereas our approach using HE allows computations to be performed independently by the nodes.

VIII. Conclusion and Future Work

In this paper, we propose DFSCC, a distributed framework for secure computing in the cloud, to enable the development of secure distributed systems. A secure distributed system developed using this framework allows for analytic tools to be implemented using HE and distributed throughout the nodes of the distributed system. Systems developed using the framework provide a high level of data security for the analytic tools since data will remain encrypted during transit to and from the cloud, and during storage and processing in the cloud. In addition, the framework provides a simple but flexible technique for sharing encrypted data among users. This approach of using HE to provide data security during data processing addresses the shortcomings of standard cryptographic schemes, and addresses some of the vulnerabilities of outsourcing data to the cloud. This will enable organizations of all types and sizes to take advantage of large pools of computing resources available in the cloud without giving up the privacy of their data.

The challenge with the existing HE schemes resides in the computation and storage overheads they incur. We addressed the computation overhead by distributing the HE computations across multiple nodes to reduce the computation time. For future work, we plan on combining the high level distribution of HE libraries and the low level parallelization of the HE operations themselves proposed in the literature. For instance, one proposed technique is to use GPGPUs to speed up the underlying operations of the libraries [31]. Combining these two approaches has the potential to significantly speed up the HE operations executed within the DFSCC framework.

References

[1] S. Kumar, F. Morstatter, and H. Liu, Twitter data analytics. Springer, 2014.

[2] A. Alexandrov et al., "The stratosphere platform for big data analytics," The VLDB Journal—The International Journal on Very Large Data Bases, vol. 23, no. 6, 2014, pp. 939–964.

[3] F. Zulkernine et al., "Towards cloud-based analytics-as-a-service (claaas) for big data analytics in the cloud," in 2013 IEEE International Congress on Big Data. IEEE, 2013, pp. 62–69.

[4] D. Talia, "Clouds for scalable big data analytics," Computer, no. 5, 2013, pp. 98–101.

[5] S. K. Sharma and X. Wang, "Live data analytics with collaborative edge and cloud processing in wireless iot networks," IEEE Access, vol. 5, 2017, pp. 4621–4635.

[6] R. Ranjan, "Streaming big data processing in datacenter clouds," IEEE Cloud Computing, vol. 1, no. 1, 2014, pp. 78–83.

[7] J. L. Asenjo et al., "Industrial data analytics in a cloud platform," Sep. 6 2016, uS Patent 9,438,648.

[8] K. Ren, C. Wang, and Q. Wang, "Security challenges for the public cloud," IEEE Internet Computing, vol. 16, no. 1, 2012, pp. 69–73.

[9] D. Zissis and D. Lekkas, "Addressing cloud computing security issues," Future Generation computer systems, vol. 28, no. 3, 2012, pp. 583–592.

[10] B. P. Rimal, E. Choi, and I. Lumb, "A taxonomy and survey of cloud computing systems," in 2009 Fifth International Joint Conference on INC, IMS and IDC. Ieee, 2009, pp. 44–51.

[11] H. Takabi, J. B. Joshi, and G.-J. Ahn, "Security and privacy challenges in cloud computing environments," IEEE Security & Privacy, vol. 8, no. 6, 2010, pp. 24–31.

[12] N. Gruschka and M. Jensen, "Attack surfaces: A taxonomy for attacks on cloud services," in 2010 IEEE 3rd international conference on cloud computing. IEEE, 2010, pp. 276–279.

[13] C. Gentry, "Fully homomorphic encryption using ideal lattices," in Proceedings of the Forty-first Annual ACM Symposium on Theory of Computing, ser. STOC '09. New York, NY, USA: ACM, 2009, pp. 169–178.

[14] A. Acar, H. Aksu, A. S. Uluagac, and M. Conti, "A survey on homomorphic encryption schemes: Theory and implementation," CoRR, vol. abs/1704.03578, 2017.

[15] M. H. Diallo, B. Hore, E. Chang, S. Mehrotra, and N. Venkatasubramanian, "Cloudprotect: Managing data privacy in cloud applications," in 2012 IEEE Fifth International Conference on Cloud Computing, June 2012, pp. 303–310.

[16] I. A. T. Hashem, I. Yaqoob, N. B. Anuar, S. Mokhtar, A. Gani, and S. U. Khan, "The rise of "big data" on cloud computing: Review and open research issues," Information systems, vol. 47, 2015, pp. 98–115.

[17] Z. Brakerski and V. Vaikuntanathan, "Fully homomorphic encryption from ring-lwe and security for key dependent messages," in Proceedings of the 31st Annual Conference on Advances in Cryptology, ser. CRYPTO'11. Berlin, Heidelberg: Springer-Verlag, 2011, pp. 505–524.

[18] S. S. Sathya, P. Vepakomma, R. Raskar, R. Ramachandra, and S. Bhattacharya, "A review of homomorphic encryption libraries for secure computation," CoRR, vol. abs/1812.02428, 2018.

[19] J. Cheon, A. Kim, M. Kim, and Y. Song, "Homomorphic encryption for arithmetic of approximate numbers," 11 2017, pp. 409–437.

[20] K. Rohloff and G. Ryan, "The palisade lattice cryptography library," 2017, retrieved: 01, 2020.

[21] J. R. Troncoso-Pastoriza, D. González-Jiménez, and F. Pérez-González, "Fully private noninteractive face verification," IEEE Transactions on Information Forensics and Security, vol. 8, no. 7, 2013, pp. 1101–1114.

[22] A. Barnett et al., "Image classification using non-linear support vector machines on encrypted data." IACR Cryptology ePrint Archive, vol. 2017, 2017, p. 857.

[23] R. Gilad-Bachrach et al., "Cryptonets: Applying neural networks to encrypted data with high throughput and accuracy," in International Conference on Machine Learning, 2016, pp. 201–210.

[24] E. Chou and Others, "Faster cryptonets: Leveraging sparsity for real-world encrypted inference," arXiv preprint arXiv:1811.09953, 2018.

[25] P. Li et al., "Multi-key privacy-preserving deep learning in cloud computing," Future Generation Computer Systems, vol. 74, 2017, pp. 76–85.

[26] P. M. Esperança, L. J. Aslett, and C. C. Holmes, "Encrypted accelerated least squares regression," 2017.

[27] Y. Aono, T. Hayashi, L. Trieu Phong, and L. Wang, "Scalable and secure logistic regression via homomorphic encryption," in Proceedings of the Sixth ACM Conference on Data and Application Security and Privacy, ser. CODASPY '16. New York, NY, USA: ACM, 2016, pp. 142–144.

[28] D. Sánchez and M. Batet, "Privacy-preserving data outsourcing in the cloud via semantic data splitting," Computer Communications, vol. 110, 2017, pp. 187–201.

[29] N. Kaaniche and M. Laurent, "Data security and privacy preservation in cloud storage environments based on cryptographic mechanisms," Computer Communications, vol. 111, 2017, pp. 120–141.

[30] B. D. Rouhani, M. S. Riazi, and F. Koushanfar, "Deepsecure: Scalable provably-secure deep learning," in Proceedings of the 55th Annual Design Automation Conference. ACM, 2018, p. 2.

[31] M. Diallo, M. August, R. Hallman, M. Kline, H. Au, and S. Slayback, Nomad: a framework for ensuring data confidentiality in mission-critical cloud-based applications, 10 2017, p. 19–44.

 ISBN: 978-1-61208-775-7

An Overview of Arithmetic Adaptations for Inference of Convolutional Neural Networks on Re-configurable Hardware

Ilkay Wunderlich
Institute of Computer Engineering
Technische Universität Dresden
Dresden, Germany
email: ilkay.wunderlich@tu-dresden.de

Benjamin Koch
AVI Systems
Freital, Germany
email: benjamin.koch@avi-systems.eu

Sven Schönfeld
AVI Systems
Freital, Germany
email: sven.schoenfeld@avi-systems.eu

***Abstract*—Convolutional Neural Networks (CNNs) have gained high popularity as a tool for computer vision tasks and for that reason are used in various applications. There are many different concepts, like single shot detectors, that have been published for detecting objects in images or video streams. However, CNNs suffer from disadvantages regarding the deployment on embedded platforms such as re-configurable hardware like Field Programmable Gate Arrays (FPGAs). Due to the high computational intensity, memory requirements and arithmetic conditions, a variety of strategies for running CNNs on FPGAs have been developed. The following methods showcase our best practice approaches for a TinyYOLOv3 detector network on a XILINX Artix-7 FPGA using techniques like fusion of batch normalization, filter pruning and post training network quantization.**

***Keywords*—convolutional neural network; image processing; re-configurable hardware; batchnorm fusing; pruning; quantization;**

I. Introduction

This section introduces the historic background of Single Shot Detectors (SSDs) and the challenges of implementing CNNs on reconfigurable hardware. Afterwards, the general "life" of neural networks is explained and expanded with the adaptation stage.

A. Single shot detector network

After the success of Convolutional Neural Networks (CNNs) for image classification, object detection stepped into the focus of research. A first brute force approach used sliding windows throughout the image with a classification network. This strategy limits itself to the granularity of the window size and window strides.

With Region-based Convolutional Neural Networks (RCNNs) a more sophisticated tool for object detection was presented. The RCNN itself contains two CNNs, which are solving the tasks of detecting objects of interest and classifying the found objects [1].

The first SSD was published by Joseph Redmon et al. with the iconic name You Only Look Once (YOLO) [2] [3]. The first version was supplemented by two updates: YOLO9000, which is also known as YOLOv2 [4] and the most recently YOLOv3 [5]. Additionally to the YOLO versions smaller versions, named TinyYOLOvX, are provided. The main focus for the following elaborations is put on the TinyYOLOv3 architecture, which optimises the trade off between detection performance and computational effort.

B. Challenges on re-configurable hardware

Implementing CNN on re-configurable hardware introduces several constraints, that affect the architecture of the used networks as well as the underlying arithmetic operations, memory access and scheduling of operations.

Using state of the art Field Programmable Gate Arrays (FPGAs), e.g., XILINX Artix-7, only fixed-point arithmetic can be implemented in an efficient way, so quantization of the network model will be necessary. Because of that, choosing quantization factors and a proper format for intermediate results to keep the deviation regarding the fixed-point model as low as possible is a key consideration to adapt a model for hardware inference.

The most challenging constraint is the limited number of logic elements to implement the building blocks of the CNN. Dissecting the parts of the neural network in candidates for hardware and software implementation is a key consideration to be made in the system architecture. Since FPGAs typically have little on-chip memory an efficient way to access memory has to be part of the architecture design as well. Also, the typical lower clock frequency in re-configurable hardware adds another trade-off.

This leads to the conclusion that a general purpose accelerator is hard to design. Every use case and neural network should be analysed according to needed performance and target platform.

C. Adaptation Stage

The "life" of any neural network can be categorized into two stages: training and inference stage. At the training stage the neural network gets trained for its later task using labelled data, which gets divided into training and test sets. A variety

 ISBN: 978-1-61208-775-7

of optimization techniques exists for the training stage. For example, one common target of optimizing the training stage aims on reducing the amount of needed iterations to reach the global minimum of the loss function. Several machine learning libraries implement optimizers for speeding up training, like Root Mean Square Propagation (RMSprop) [6] or Adaptive Moment Estimation (ADAM) [7].

The inference stage is the application of the neural network, after it is properly trained. Adaptations to the network might be needed, depending on the hardware platform on which the inference stage takes place. The entirety of the adaptation work flow is summarized as the adaptation stage.

In section II and III, two optional but very useful adaptation steps are introduced. In Section IV, the mandatory adaptation of switching the arithmetic backbone of the network from floating point to fix point operations is presented. The benefits of these techniques are summarized in section V.

II. Batchnorm Fusing

In this section, the concept of batch normalization and the general layout of a convolution layer are briefly explained. Thereafter, formulae and results of eliminating the batch normalization layer are given.

A. Background of Batch Normalization

Batch normalization [8], which is often times abbreviated with batchnorm, is a sub layer used to reduce internal covariate shifts. These shifts are defined as changes in the distribution of the network's activation, which are caused by changes in the parameters of the network during training stage. Diminishing the covariate shift enables higher learning rates, reduces the risk of getting stuck in poor local minima and prevents vanishing or exploding gradients. Another advantageous side effect of batch normalization is the increased generalization ability of the network [8]. Many modern networks are using batchnorm sub layers, e.g., YOLOv3 [5], MobileNet [9] and ResNet [10].

The batch normalization sub layer is located between the convolutional layer and the activation sub layer which is illustrated for layer i in Figure 1.

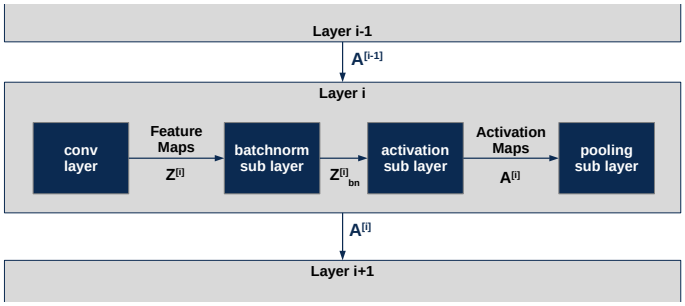

Fig. 1. Example of a general convolutional layer (abbreviated with conv) with its subsequent batchnorm, activation and pooling sub layer and their output descriptions.

This sub layer contains a set of up to four parameters:

1) **Mini-Batch Mean**:
The mini-batch mean $\mu^{[i]}$ is measured regarding the mean of the feature maps of the current mini-batch $\mathcal{B}^{[i]} = \left\{z_1^{[i]}, ..., z_m^{[i]}\right\}$ at each batchnorm sub layer i in the network. This metric gets updated for each mini-batch and epoch during the training stage:

$$\mu^{[i]} \leftarrow \frac{1}{m} \cdot \sum_{j=0}^{m} \mathcal{B}^{[i]} \tag{1}$$

2) **Mini-Batch Variance**:
Similar to mini-batch mean with respect to the variance of the current mini-batch:

$$\sigma^{2^{[i]}} \leftarrow \frac{1}{m} \cdot \sum_{j=0}^{m} (\mathcal{B}^{[i]} - \mu^{[i]})^2 \tag{2}$$

3) **Scale**:
Trained scaling term $\gamma^{[i]}$, which is an optional and trainable component of the batchnorm sub layer.

4) **Shift**:
Trained shifting term $\beta^{[i]}$, which is an optional and trainable component of the batchnorm sub layer as well. Commonly used instead of bias parameters b in the convolutional layer.

The batchnorm parameters are one dimensional vectors, which are applied to each feature map from the previous convolutional layer separately. The amount of elements is determined by the amount of filters used in the conv layer. The forward propagation step for the batchnorm sub layer is given in (3):

$$Z_{bn}^{[i]} = BN(Z^{[i]}) = \gamma^{[i]} \cdot \underbrace{\frac{Z^{[i]} - \mu^{[i]}}{\sqrt{\sigma^{2^{[i]}} + \epsilon}}}_{normalization} + \beta^{[i]} \tag{3}$$

where ϵ is a small scalar value added to the variance to provide numerical stability (e.g., keras default: $\epsilon = 0.001$ [11]).

B. Fusion of Batchnorm Parameters into Convolutional Parameters

In order to get rid of the computational effort of the batch normalization sub layer, it is recommended to fuse the batchnorm parameters into the convolutional parameters before entering the inference stage. To derive the formulae for batchnorm fusing, the kernel convolution, which is performed filter-wise in the convolutional layer is introduced for layer i:

$$Z^{[i]} = Conv\left(A^{[i-1]}; W^{[i]}, b^{[i]}\right) = A^{[i-1]} * W^{[i]} + b^{[i]} \tag{4}$$

where $A^{[i-1]}$ denotes the three dimensional activation map from the previous layer of matrix shape (w, h, c) - (width,height,color channels). $W^{[i]}$ is the filter matrix of shape (f_w, f_h, c) and $b^{[i]}$ represents the optional bias vector is shape $(c, 1)$. For easier reading, the layer indices $()^{[i]}$ are skipped in the following arguments with hinting $A^{[i-1]}$ as A_{prev}. It can be shown, that the convolution operation $Conv(A; W, b)$ holds the following property:

$$k \cdot Conv\left(A_{prev}; W, b\right) + h = Conv\left(A_{prev}; k \cdot W, k \cdot b + h\right) \tag{5}$$

for $k, h = const.$ and $k, h \in \mathbb{R}^c$. With (3),(4) and (5) the formulae for batchnorm fusing are derived as follows:

Replacing $Z^{[i]}$ from (3) with (4):

$$Z_{bn} = \gamma \cdot \frac{Conv\left(A_{prev}; W, b\right) - \mu}{\sqrt{\sigma^2 + \epsilon}} + \beta \tag{6}$$

Reordering to get the equation above to a similar form as shown in (5):

$$Z_{bn} = \underbrace{\frac{\gamma}{\sqrt{\sigma^2 + \epsilon}}}_{k} \cdot Conv\left(A_{prev}; W, b\right) + \underbrace{\beta - \frac{\gamma \cdot \mu}{\sqrt{\sigma^2 + \epsilon}}}_{h} \tag{7}$$

which can be written as:

$$Z_{bn} = Conv\left(A_{prev}; \tfrac{\gamma}{\sqrt{\sigma^2+\epsilon}} \cdot W, \tfrac{\gamma}{\sqrt{\sigma^2+\epsilon}} \cdot b + \beta - \tfrac{\gamma \cdot \mu}{\sqrt{\sigma^2+\epsilon}}\right) = \tag{8}$$

$$Conv\left(A_{prev}; \frac{\gamma}{\sqrt{\sigma^2 + \epsilon}} \cdot W, \frac{\gamma}{\sqrt{\sigma^2 + \epsilon}} \cdot (b - \mu) + \beta\right)$$

With (8) the formulae for the fused parameters W_{bn}, b_{bn} are derived as:

$$W_{bn} = \frac{\gamma}{\sqrt{\sigma^2 + \epsilon}} \cdot W \tag{9}$$

$$b_{bn} = \frac{\gamma}{\sqrt{\sigma^2 + \epsilon}} \cdot (b - \mu) + \beta \tag{10}$$

$$Z_{bn} = Conv\left(A_{prev}; W_{bn}, b_{bn}\right) \tag{11}$$

For convolution layers trained without biases (10) is reduced to:

$$b_{bn} = \beta - \frac{\gamma \cdot \mu}{\sqrt{\sigma^2 + \epsilon}} \tag{12}$$

C. Benefits of Batchnorm fusing

The emphasis for estimating the benefits of batchnorm fusing is shown with respect to the reduction of Floating Point Operations (FLOPS). In Figure 2, the amount of needed FLOPS for the first layers of a TinyYOLOv3 network without batchnorm fusing for an input image size of $416 \times 416 \times 3$ (width, height, color channels) is shown. This network requires approximately 5.5 GFLOPS for processing one image from the input layer to the output layers, excluding the processing steps in the YOLO back-end. After fusing the batchnorm parameters into the convolutional parameters, the FLOPS count is reduced by 23.8 MFLOPS.

This decrease sounds low with a reduction factor of only 0.4%. But more important is the avoidance of batchnorm sub layers as a whole, because each sub layer requires additional logic elements, complexity in the control loop and power. Another benefit is the preparation for pruning, which is performed on the fused weight matrices.

III. Filter Pruning

The general concept of pruning and the proposed routine are presented. The results of pruning are displayed using parameter count and FLOPS as optimization target.

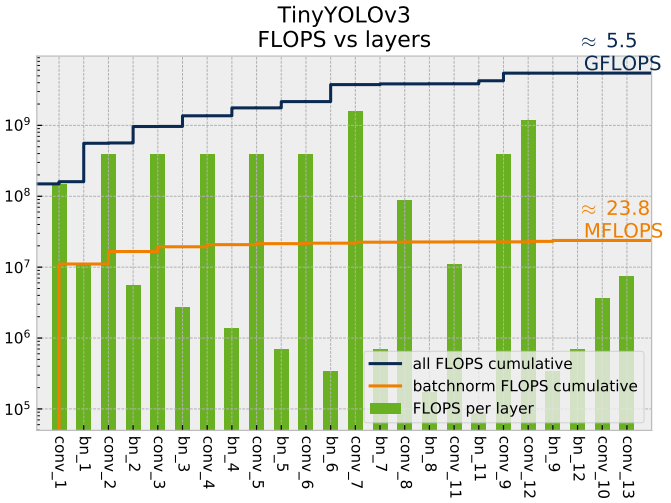

Fig. 2. Required FLOPS (y-axis, logarithmic) for all convolutional layer with their batchnorm sub layer (x-axis) for the TinyYOLOv3 architecture.

A. Pruning Background

In contrast to batch normalization being introduced in 2015, research on pruning of neural networks already started at the end of 1980s [12] and the beginning of 1990s [13]. The goal of pruning is reducing the network size by removing redundant connections while maintaining the performance of the network. M.C. Mozer and P. Smolensky pictorially call this technique as “trimming the fat from a network” [13]. In order to determine redundant connections several metrics are proposed in literature [14]. In the following, the emphasis is put on CNNs and particularly on pruning whole filters after batchnorm fusing.

B. Pruning Metrics

In this section, two metrics for determining filter candidates suitable for pruning are introduced. These metrics help finding filters f_n of the weight matrix W, which have no or low impact for the forward propagation. The total amount of filters stored in the weight matrix W is denoted with nf. A trivial example for a prunable filter is one which coefficients are all zero ($f_n = \mathbf{0}$).

1) The **Frobenius Norm** for filter f_n is defined as the following scalar value:

$$||f_n||_F = \sqrt{\sum_{w,h,c} \left(f_n[w,h,c]\right)^2} \tag{13}$$

This definition gets expanded for the whole weight matrix by stacking the norms of the all filters f_n to one vector:

$$||W||_F = \left[||f_0||_F, ..., ||f_n||_F, ..., ||f_{nf}||_F\right]^T \tag{14}$$

2) The **filter sparsity** is a metric for determining the sparsity of a filter. It is defined as the percentage of values close to zero of a filter f_n:

$$Sp_\epsilon(f_n) = 1 - \frac{\mathcal{C}(|f_n| < \epsilon)}{\mathcal{C}(f_n)} \quad (15)$$

where $\mathcal{C}(f_n)$ denotes the cardinality of the filter f_n (16) and $\mathcal{C}(|f_n| < \epsilon)$ the conditional cardinality of f_n (17).

$$\mathcal{C}(f_n) = \sum_{w,h,c} 1 \quad (16)$$

$$\mathcal{C}(|f_n| < \epsilon) = \sum_{w,h,c} \begin{cases} 1, & |f[w,h,c]| < \epsilon \\ 0, & |f[w,h,c]| \geq \epsilon \end{cases} \quad (17)$$

Similarly to (14) this equation is expanded to:

$$Sp_\epsilon(W) = [Sp_\epsilon(f_0), ..., Sp_\epsilon(f_n), ..., Sp_\epsilon(f_nf)]^T \quad (18)$$

C. Pruning Routine

The previously mentioned metrics are used in the proposed pruning routine, which is based on the following inputs:

- The maximum deviation of Mean Average Precision (MAP): Δ_{MAP}.
- A representative pruning data set to continuously calculate the mean average precision.
- An optional Starting threshold T_{start}, which is 0 by default.
- A value by which the threshold gets incremented δ_T: E.g., $\delta_T = 0.01$.

The pruning routine determines the chosen metric for every filter in the CNN, as well as the initial MAP of the network on the pruning data set beforehand. Afterwards every filter, which is below the threshold T, is removed and the MAP is calculated again. As long as deviation is lower than the maximum deviation Δ_{MAP}, the threshold is incremented by δ_T. This procedure is repeated until Δ_{MAP} is reached.

After the pruning routine is finished, the possibility of additional training in order to slightly fine tune the remaining filters to reach the initial MAP is possible. It is advisable to perform the fine tuning with a low learning rate and early stopping to avoid over fitting of the network to the pruning data set.

D. Pruning Result

The following example is based on a TinyYOLOv3 network, which is trained on the TU Darmstadt Pedestrian Dataset [15]. The original model, fused model, and the pruned example models as well as additional information are provided in a separate GitHub repository [16].

The maximum deviation of MAP is set to $\Delta_{mAP} = 1\%$ and the threshold increment $\delta_T = 0.02$. In Figure 3, the results of the pruning routine are shown using Frobenius Norm $||W||_F$ and filter sparsity $Sp_{\epsilon=0.003}(f_n)$ as the pruning metric. The results cover the total parameter and filter count of the CNN.

It is notable that the parameter count decrease is higher than the decrease in filters. The reason is, that the higher the amount of filters in a layer i the more likely is to encounter prunable filters. The amount of parameters stored in such layers are way larger, because the previous layers $i-1$ typically have higher filter counts as well. This behaviour is visualized for TinyYOLOv3 in Figure 4 by plotting the percentage of stored parameters per layer to the overall TinyYOLOv3 parameter count.

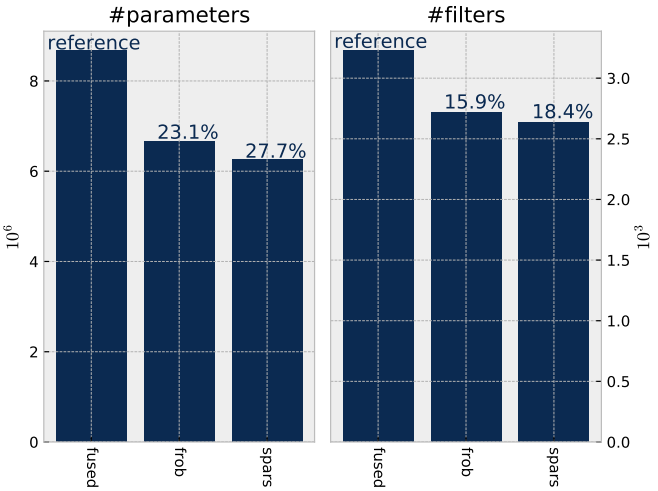

Fig. 3. Pruning result showing for the TinyYOLOv3 network example comparing fused model (fused) and pruned model with Frobenius Norm (fro) and Filter Sparsity (spars) as metric. Top of bar: reduction percentage or reference to it. Left: total parameter count. Right: filter count of the network.

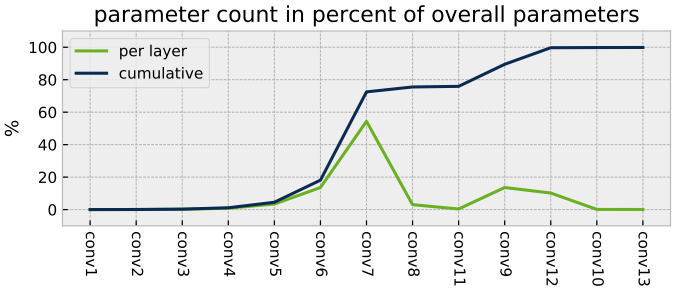

Fig. 4. Percentage of parameters for each layer. Per layer and cumulative.

With the reduction of the parameter count of approximately 23.1% for Frobenius and 27.7% for sparsity pruning a higher reduction of FLOPS can be observed compared to batchnorm fusing from Subsection II-C. The comparison of FLOPS reduction percentages between original model (with batchnorm sub layers), fused model and pruned models is visualized in Figure 5. The reduction of FLOPS is 13.3% for Frobenius and 15.7% for sparsity pruning compared to the original model with batchnorm sub layers.

Conclusively, it has to be stated that a very high pruning result (e.g., $> 80\%$ of parameters pruned) needs more investigation. The CNN might either be oversized or not properly trained. In both cases, the filter weights are still close to the initialization. This leads to low values for the introduced metrics, which result in a huge amount of prunable filters.

 ISBN: 978-1-61208-775-7

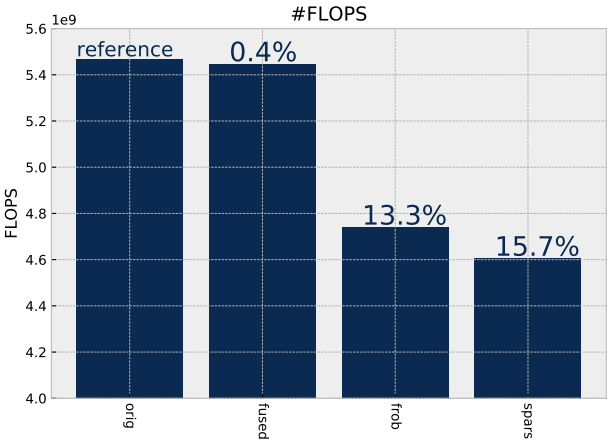

Fig. 5. FLOPS comparison (y-axis) for the TinyYOLOv3 network example. Reduction percentages (top of bar) using the "unfused" original model (orig) as reference.

IV. Quantization

In this section, a method of changing the arithmetic backbone from floating point to integer arithmetic is given and evaluated with investigating on the deviation between floating point CNN and integer CNN. In conclusion, other methods are mentioned and briefly explained.

A. Background

Training for CNNs is traditionally performed using 32 bit floating point arithmetic. The main reason is, that calculating the gradient during CNN training requires high value resolution for which integer or fixed-point systems are insufficient. However, using floating point arithmetic on re-configurable hardware such as FPGAs for CNN inference comes with plenty of disadvantages: higher computation effort, more memory storage and increase of bus widths and counts. These disadvantages can be solved by using a quantized twin of the CNN with a minimal or no drop-off in inference accuracy.

B. Quantization Approach

A straight forward approach based on scaling the floating point values is proposed. This is realized by determining a positive whole-numbered scaling factor S, e.g., $S = 256$. It is advisable to select this factor from the set of powers of two $S = 2^P$ where $P \in \mathbb{N}$, because integer division by a power of two becomes a right shift by the power of two P:

$$\frac{X}{S} = \frac{X}{2^P} = Rshift(X, P) \quad P \in \mathbb{N} \tag{19}$$

This is especially useful since the integer product $Z = MUL_S(X_1, X_2)$ of two mapped operands $X_i = x_i \cdot S$ requires a division by the scaling factor S to obtain $Z = S \cdot x_1 \cdot x_2 + e = S \cdot z + e$ where e denotes the quantization error:

$$\underbrace{Mul_S(X,Y)}_{=:Z} = \frac{X \cdot Y}{S} \overset{S=2^P}{=\!=} Rshift(X \cdot Y, P) = S \cdot \underbrace{x \cdot y}_{=:z} + e \tag{20}$$

Using shift operations instead of divisions spares hardware resources and reduces the combinatorial path, allowing a higher clock frequency of the hardware. With this background, the quantization approach can be introduced:

- **parameter and input quantization:**
  Every parameter value V (floating point, usually 32 bit) of the CNN is quantized by multiplying with the scaling factor S and rounding to integer values. For this, the integer format $int16$ is used.

$$V_{quant} = int16(round(V * S)) \tag{21}$$

  The input $A^{[0]}$ is quantized similarly for every pixel value $V \in A^{[0]}$.
- **quantized convolution:**
  The forward propagation of the convolution layer, which is split up into the filter convolution $Conv$ and the bias addition Add, is adjusted by appending a right shift operator $Rshift$ after the convolution step as illustrated in Figure 6. The convolution itself is performed using

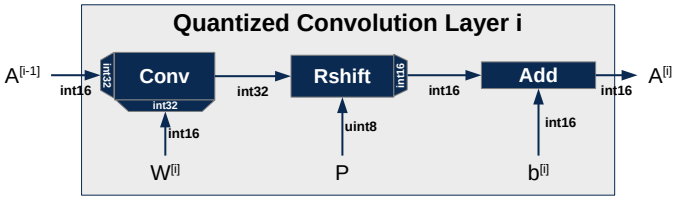

Fig. 6. Quantized Convolution Layer i without pooling and activation sub layer.

  $int32$ range with expansion of the previous activation $A^{[i-1]}$ and the filter parameters $W^{[i]}$ to $int32$:

$$A^{[i-1]} \leftarrow int32(A^{[i-1]})$$
$$W^{[i]} \leftarrow int32(W^{[i]})$$
$$A^{[i]} \leftarrow Conv(A^{[i-1]}, W^{[i]})$$

  After the right shift by P the format is reduced to $int16$ and the bias $b^{[i]}$ is added:

$$A^{[i]} \leftarrow Rshift(A^{[i]}, P)$$
$$A^{[i]} \leftarrow int16(A^{[i]})$$
$$A^{[i]} \leftarrow Add(A^{[i]}, b^{[i]})$$

- **quantized activation function:**
  An approved activation function for CNNs is the Leaky Rectified Linear Unit (ReLU$_\alpha$) as defined in (22) and shown in Figure 7.

$$a = ReLU_\alpha(z) = \begin{cases} z, & z > 0 \\ \alpha \cdot z, & z \leq 0 \end{cases} \tag{22}$$

  The value of the subscripted α is denoting the negative slope. Common values for α are 0.01, 0.2 and 0.3 (default values of Caffe2, Tensorflow and Keras). For $\alpha = 0$ the Rectified Linear Unit (ReLU) without any "leakage" is obtained.

 ISBN: 978-1-61208-775-7

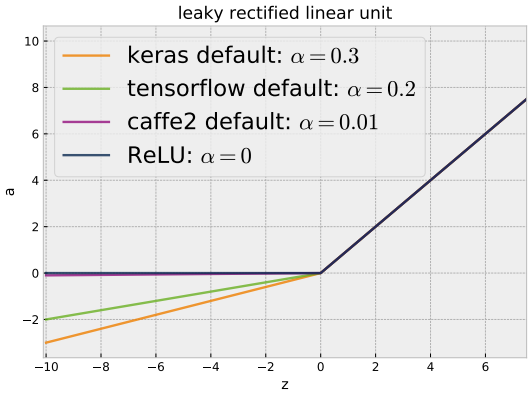

Fig. 7. ReLU$_\alpha$ for various α values.

ReLU$_\alpha$ is a suitable activation function for targeting embedded systems, since its computational effort compared to other activation functions like hyperbolic tangent $tanh$ or sigmoid σ is low. In order to make it even more suitable, the negative slope α is chosen from the set of negative powers of two: $\alpha = 2^{-P_\alpha}$ where $P_\alpha \in \mathbb{N}$: E.g., $\alpha = 2^{-4} = 0.0625$.
Analogously to the convolution quantization the multiplication by α is rewritten using the right shift operator $Rshift$ and the relationship from (19):

$$\alpha \cdot z = 2^{-P_\alpha} \cdot z = \frac{z}{2^{P_\alpha}} = Rshift(z, P_\alpha) \tag{23}$$

This simplifies (22) for integer arithmetic as follows:

$$a = ReLU_\alpha(z) = \begin{cases} z, & z > 0 \\ Rshift(z, P_\alpha), & z \leq 0 \end{cases} \tag{24}$$

- **pooling:**
  No adaptations are needed for transferring max pooling to integer arithmetic. For average pooling, further right shift simplifications can be made.

C. Quantization Deviation

The deviation between the trained floating point model and its quantized integer twin is estimated by the outputs of each convolution layer and the following sub layers (activation, pooling). The deviation is calculated using the Mean Squared Error (MSE) as metric:

$$MSE(A^{[i]}_{float}, A^{[i]}_{int}) = \frac{1}{N} \sum_{\substack{v \in A^{[i]}_{float} \\ w \in A^{[i]}_{int}}} \left(v - \frac{w}{S}\right)^2 \tag{25}$$

with N denoting the number of elements (cardinality) of the compared arrays:

$$N = \mathcal{C}\left(A^{[i]}_{float}\right) = \mathcal{C}\left(A^{[i]}_{int}\right) \tag{26}$$

with S being the scaling factor used for quantization.

Figure 8 shows the layer-wise calculated MSE between floating point model and integer model using a scaling factor $S = 2^8 = 256$. The used network is the pruned version of the pedestrian detector network introduced in Subsection III-D.

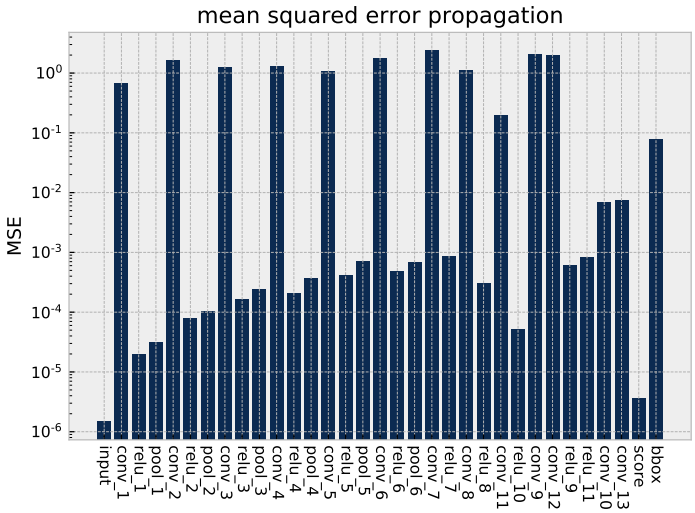

Fig. 8. MSE (y-axis, logarithmic) per layer (x-axis) for TinyYOLOv3 using a scaling factor of $S = 256$ on a randomly chosen image from the test set of TU Darmstadt Pedestrian Dataset [15].

A slow increase of the MSE is observable regarding the outputs of each TinyYOLOv3 layer. This increase plateaus around the fifth layer and stays below a MSE of 0.001. The ReLU$_\alpha$ activation function helps dealing with deviation from the convolution operation by dimming out negative deviation due to the lower negative slope. But most significantly is that the final convolution layers $conv\_10$ and $conv\_13$ display low MSEs. This results in a neglectable difference at the detected classes scores $score$ of $\Delta_{score} = 0.0019$ and a deviation of maximum 2 pixel for the found Bounding Box (bbox).

Similar behaviors with low or neglectable deviations are observed for other TinyYOLOv3 networks, which are trained for different detection tasks, as well as other architectures such as Visual Geometry Group (VGG) classification networks [17].

D. Advanced Approaches

In the following, other promising strategies and more advanced approaches are briefly described:

One approach explicitly tackles the size of the network compressing them by factors up to 30 using weight sharing via k-means clustering method and code books with HUFFMAN encoding [18].

Another goal is to reduce the internal bit widths. In order to achieve that, a distribution based quantization scheme is developed quantizing each filter accordingly to the weight distribution. With that, lower bit widths like 4 bit weights and 8 bit activations are possible [19].

Structural adaptations in architecture designs are proposed for example with MobileNet [9]. MobileNet introduces a new layer type: the depth-wise convolution layer. One of its advantages compared to the "classic" convolution layer is the lower amount of FLOPS required for inference. In MobileNet the ReLU function is extended with a threshold parameter θ

 ISBN: 978-1-61208-775-7

for preventing overflows and allowing lower bit widths. The default value used in MobileNet for θ is 6. The so called ReLU-6 function is directly supported in tensorflow [20].

Another very promising idea is based on replacing multiplications with XNOR operations, which is speeding up inference heavily [21]. The so called XNOR-Net is created for image classification using binary activation and adapted gradient descent methods for training. However, its main disadvantage is the lack of framework support for the commonly used machine learning frameworks.

In order to extend the existing design, appropriate concepts of the presented advanced approaches will be considered.

V. Conclusion

The work flow of the adaptation stage with the following methods is shown:

- **Fusion of batch normalization sub layers** (Section II): Formulae and benefits for eliminating the batchnorm sub layers by fusing the batchnorm parameters into the weight and bias parameters of the preceding convolutional layer are presented.
- **Pruning of convolutional filters** (Section III): A pruning routine with two different pruning metrics is elaborated. The advantages and reduction potentials are exemplary stated.
- **Creation of a quantized twin using integer arithmetic** (Section IV): A straight forward approach with utilisation of shift operation to speed up inference is given. More advanced strategies are mentioned and will be investigated in future works.

Overall, the goal of running a TinyYOLOv3 CNN architecture on an Artix-7 FPGA is accomplished. In order to speed up inference on the FPGA, more optimizations will be developed and supplemented with fitting improvements from other strategies as presented in Subsection IV-D.

Acknowledgment

The investigation of the described adaptations for CNNs on re-configurable hardware are developed at AVI Systems at Freital Germany as well as the chair of VLSI-Design, Diagnostic and Architecture of the Institute of Computer Engineering at Technische Universität Dresden.

References

[1] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,”, Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 580–587, 2014.

[2] J. Redmon and A. Angelova, “Real-time grasp detection using convolutional neural networks,” IEEE International Conference on Robotics and Automation (ICRA), pp. 1316–1322, 2015.

[3] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You Only Look Once: Unified, real-time object detection,” IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 779–788, 2016.

[4] J. Redmon and A. Farhadi, “YOLO9000: Better, faster, stronger,” IEEE Conference on Computer Vision and Pattern Recognition (CVPR), doi. 10.1109/cvpr.2017.690, 2017.

[5] J. Redmon and A. Farhadi, “YOLOv3: An incremental improvement,”, arXiv:1804.02767, 2018.

[6] T. Tieleman and G. Hinton, “Divide the gradient by a running average of its recent magnitude,” Tech. Rep., Technical report, p. 31, 2012.

[7] D.P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” arXiv:1412.6980, 2014.

[8] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” Proceedings of the 32nd International Conference on Machine Learning, pp. 448–456, 2015.

[9] A.G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T.Weyand, et al., “MobileNets: Efficient convolutional neural networks for mobile vision applications,” arXiv:1704.04861, 2017.

[10] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” IEEE Conference on Computer Vision and Pattern Recognition (CVPR), doi. 10.1109/cvpr.2016.90, 2016.

[11] F. Chollet and others, “BatchNormalization,” https://keras.io/layers/normalization/, retrieved: 01.2020, 2014.

[12] M. C. Mozer and P. Smolensky, “Skeletonization: A technique for trimming the fat from a network via relevance assessment,” Advances in Neural Information Processing, pp. 107–115, 1989.

[13] E. D. Karnin, “A simple procedure for pruning back-propagation trained neural networks,” IEEE transactions on neural networks, vol. 1, no. 2, pp. 239–242, 1990.

[14] R. Reed, “Pruning algorithms-a survey,” IEEE transactions on neural networks, vol. 4, no. 5, pp. 740–747, 1993.

[15] M. Andriluka, S. Roth, and B. Schiele, “People-tracking-by-detection and people-detection-by-tracking,” IEEE Conference on computer vision and pattern recognition, pp. 1–8, June 2008.

[16] I. Wunderlich, “CNN4FPGA,” GitHub repository, https://github.com/IlkayW/CNN4FPGA.git, 2020.

[17] K. Simonyan and A. Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition,” arXiv:1409.1556, 2014.

[18] S. Han, H. Mao, and W. J. Dally, “Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding,” arXiv:1510.00149, 2015.

[19] S. Sasaki, A. Maki, D. Miyashita, and J. Deguchi, “Post training weight compression with distribution-based filter-wise quantization step,” IEEE Symposium in Low-Power and High-Speed Chips (COOL CHIPS), pp. 1–3 , 2019.

[20] Tensorflow “ReLU-6,”, https://www.tensorflow.org/api_docs/python /tf/nn/relu6?version=stable, retrieved: 01.2020.

[21] M. Rastegari, V. Ordonez, J. Redmon, and A. Farhadi, “XNOR-Net: ImageNet classification using binary convolutional neural networks,” Lecture Notes in Computer Science, pp. 525–-542, doi. 10.1007/978-3-319-46493-0_32, 2016.

 ISBN: 978-1-61208-775-7