PATTERNS 2021

The Thirteenth International Conferences on Pervasive Patterns and Applications

ISBN: 978-1-61208-850-1

April 18 - 22, 2021

PATTERNS 2021 Editors

Herwig Mannaert, University of Antwerp, Belgium

Cosmin Dini, IARIA, EU/USA

PATTERNS 2021

Forward

The Thirteenth International Conferences on Pervasive Patterns and Applications (PATTERNS 2021), held on April 18 - 22, 2021, continued a series of events targeting the application of advanced patterns, at-large. In addition to support for patterns and pattern processing, special categories of patterns covering ubiquity, software, security, communications, discovery and decision were considered. It is believed that patterns play an important role on cognition, automation, and service computation and orchestration areas. Antipatterns come as a normal output as needed lessons learned.

The conference had the following tracks:

- Patterns basics
- Patterns at work
- Discovery and decision patterns

Similar to the previous edition, this event attracted excellent contributions and active participation from all over the world. We were very pleased to receive top quality contributions.

We take here the opportunity to warmly thank all the members of the PATTERNS 2021 technical program committee, as well as the numerous reviewers. The creation of a high quality conference program would not have been possible without their involvement. We also kindly thank all the authors that dedicated much of their time and effort to contribute to PATTERNS 2021. We truly believe that, thanks to all these efforts, the final conference program consisted of top quality contributions.

Also, this event could not have been a reality without the support of many individuals, organizations and sponsors. We also gratefully thank the members of the PATTERNS 2021 organizing committee for their help in handling the logistics and for their work that made this professional meeting a success.

We hope PATTERNS 2021 was a successful international forum for the exchange of ideas and results between academia and industry and to promote further progress in the area of pervasive patterns and applications.

PATTERNS 2021 Steering Committee

Herwig Manaert, University of Antwerp, Belgium
Wladyslaw Homenda, Warsaw University of Technology, Poland
Patrick Siarry, Université Paris-Est Créteil, France
Yuji Iwahori, Chubu University, Japan
Alexander Mirnig, University of Salzburg, Austria
Adel Al-Jumaily, University of Technology, Australia
George A. Papakostas, International Hellenic University – Kavala, Greece

PATTERNS 2021 Publicity Chair

Jose Luis García, Universitat Politecnica de Valencia, Spain
Lorena Parra, Universitat Politecnica de Valencia, Spain

PATTERNS 2021 Industry/Research Advisory Committee

Christian Kohls, TH Köln, Germany

PATTERNS 2021

Committee

PATTERNS 2021 Steering Committee

Herwig Manaert, University of Antwerp, Belgium
Wladyslaw Homenda, Warsaw University of Technology, Poland
Patrick Siarry, Université Paris-Est Créteil, France
Yuji Iwahori, Chubu University, Japan
Alexander Mirnig, University of Salzburg, Austria
Adel Al-Jumaily, University of Technology, Australia
George A. Papakostas, International Hellenic University – Kavala, Greece

PATTERNS 2021 Industry/Research Advisory Committee

Christian Kohls, TH Köln, Germany

PATTERNS 2021 Publicity Chairs

Jose Luis García, Universitat Politecnica de Valencia, Spain
Lorena Parra, Universitat Politecnica de Valencia, Spain

PATTERNS 2021 Technical Program Committee

Andrea F. Abate, University of Salerno, Italy
Akshay Agarwal, IIIT Delhi, India
Adel Al-Jumaily, University of Technology, Australia
Ali Reza Alaei, School of Business and Tourism, Australia
Sidnei Alves De Araujo, Nove de Julho University (UNINOVE), Sao Paulo, Brazil
Danilo Avola, Sapienza University of Rome, Italy
Johanna Barzen, University of Stuttgart, Germany
Nadjia Benblidia, Saad Dahlab University - Blida1, Algeria
Anna Berlino, Consultant in Tourism Sciences and Valorization of Cultural and Tourism Systems, Italy
Uwe Breitenbücher, IAAS - University of Stuttgart, Germany
Alceu S. Britto, Pontifical Catholic University of Paranā (PUCPR), Brazil
Jean-Christophe Burie, L3i laboratory | La Rochelle University, France
Simone Cammarasana, CNR-IMATI, Genova, Italy
David Cárdenas-Peña, Universidad Tecnológica de Pereira, Colombia
Bidyut B. Chaudhuri, Indian Statistical Institute, India
Sneha Chaudhari, AI Organization | LinkedIn, USA
Diego Collazos, Universidad Nacional de Colombia sede Manizales, Colombia
Sergio Cruces, University of Seville, Spain
Mohamed Daoudi, Institut Mines-Telecom / Telecom Lille, France

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

Usage of Iterated Local Search to Improve Firewall Evolvability

Haerens Geert
Antwerp University, Engie
Brussels, Belgium
email: geert.haerens@engie.com

***Abstract*—The Transmission Control Protocol/Internet Protocol (TCP/IP) based firewall is a notorious non-evolvable system. Changes to the firewall often result in unforeseen side effects, resulting in the unavailability of network resources. The root cause of these issues lies in the order sensitivity of the rule base and hidden relationships between rules. It is not only essential to define the correct rule. The rule must be placed at the right location in the rule base. As the rule base becomes more extensive, the problem increases. According to Normalized Systems, this is a Combinatorial Effect. In previous research, an artifact has been proposed to build a rule base from scratch in such a way that the rules will be disjoint from each other. Having disjoint rules is the necessary condition to eliminate the order sensitivity and thus the evolvability issues. In this paper, an algorithm, based on the Iterated Local Search metaheuristic, will be presented that will disentangle the service component in an existing rule base into disjoint service definitions. Such disentanglement is a necessary condition to transform a non-disjoint rule base into a disjoint rule base.**

Keywords–Firewall; Rule Base; Evolvability; Metaheuristic; Iterated Local Search.

I. INTRODUCTION

The TCP/IP based firewall has been and will continue to be an essential network security component in protecting network-connected resources from unwanted traffic. The increasing size of corporate networks and connectivity needs has resulted in firewall rule bases increasing considerably. Large rule bases have a nasty side effect. It becomes increasingly difficult to add the right rule at the correct location in the firewall. Anomalies start appearing in the rule base, resulting in the erosion of the firewall's security policy or incorrect functioning. Making changes to the firewall rule base becomes more complex as the size of the system grows. An observation shared by Forrester [1] and the firewall security industry [2] [3]. A more detailed literature review on the topic can be found in [4].

Normalized Systems (NS) theory [5] defines a Combinatorial Effect (CE) as the effect that occurs when the impact of a change is proportional to the nature of the change and the system's size. According to NS, a system that suffers from CE is considered unstable under change or non evolvable. A firewall suffers from CE. The evolvability issues are the root cause of the growing complexity of the firewall as time goes by.

The order sensitivity plays a vital role in the evolvability issues of the rule base. The necessary condition to remove the order sensitivity is known, being non-overlapping or disjoint rules. However, firewall rule bases don't enforce that condition, leaving the door open for misconfiguration. While previous work investigates the causes of anomalies [6] [7], detecting anomalies [8] [9] [10] and correcting anomalies at the time of entering new rules in the rule base [8], to the best of our knowledge and efforts, no work was found that tries to construct a rule base with ex-ante proven evolvability (= free of CE). While previous methods are reactive, this paper proposes a proactive approach.

Issues with evolvability of the firewall rule base induce business risks. The first is the risk of technical communication paths not being available to execute business activities properly. The second is that flaws in the rule base may result in security issues, making the business vulnerable for malicious hacks resulting in business activities' impediment.

In this paper, we propose an artifact, an algorithm, that aims at converting an existing non-evolvable rule base into an evolvable rule base. Design Science [11] [12] is suited for research that wants to improve things through artifacts (tools, methods, algorithms, etc.). The Design Science Framework (see Figure 1) defines a relevance cycle (solve a real and relevant problem) and rigor cycle (grounded approach, usage of existing knowledge and methodologies).

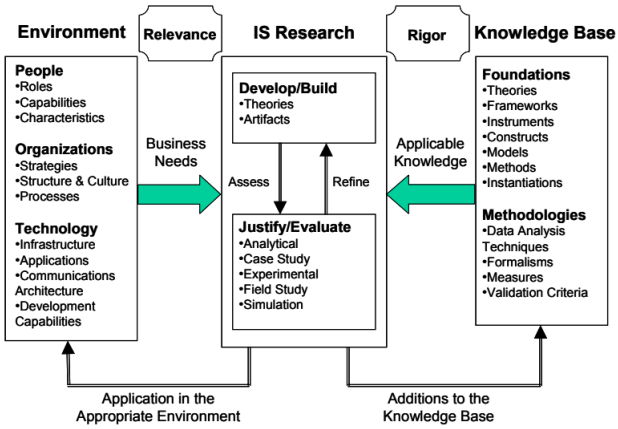


Figure 1. The Design Science Framework - from [11] .

The Design Science Process (see Figure 2) guides the artifact creation process according to the relevance and rigor cycle. What follows is structured according to the Design Science process.

Section II introduces the basic concepts of firewalls, fire-

 ISBN: 978-1-61208-850-1

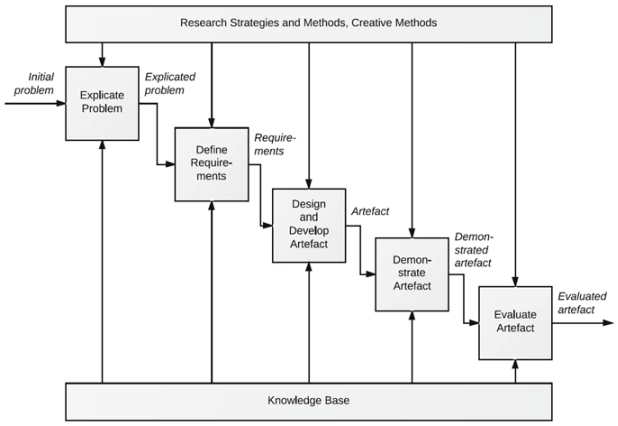


Figure 2. The Design Science Process - from [12].

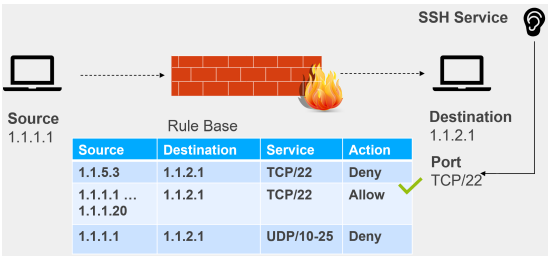


Figure 3. Firewall concepts.

wall rule relationships, Normalized Systems, and the evolvability issues of the firewall. In Section III, we will discuss the requirements for an algorithm that will transform a non-evolvable rule base, into an evolvable rule base. Section IV will build the different components of the proposed algorithm using the Iterated Local Search metaheuristic. In Section V, the algorithm will be demonstrated in a number of cases. In Section VI, we evaluate and discuss our findings and we wrap-up with a conclusion in Section VII.

II. Problem Description

The first part of this section will explain how a firewall works and the concept of firewall group objects. The second part will discuss the relationships between firewall rules and introduces the Normalized Systems theory.

A. Firewall basics

An Internet Protocol Version 4 (IP4) TCP/IP based firewall, located in the network path between resources, can filter traffic between the resources, based on the Layer 3 (IP address) and Layer 4 (TCP/UDP ports) properties of those resources [13]. UDP stands for User Datagram Protocol and is, next to TCP, a post based communcation protocol at the 4th level of the Open Systems Interconnection Model (OSI Model) [14]. Filtering happens by making use of rules. A rule is a tuple containing the following elements: <Source IP, Destination IP, Protocol, Destination Port, Action>. IP stands for IP address and is a 32-bit number that uniquely identifies a networked resource on a TCP/IP based network. The protocol can be TCP or UDP. Port is a 16-bit number (0 - 65.535) representing the TCP or UDP port on which a service is listening on the 4th layer of the OSI-stack. When a firewall sees traffic coming from a resource with IP address =<Source IP>, going to resource =<Destination IP>, addressing a service listening on Port = <Destination port>, using Protocol = <Protocol>, the firewall will look for the first rule in the rule base that matches Source IP, Destination IP, Protocol and Destination Port, and will perform an action = <Action>, as described in the matched rule. The action can be “Allow” or “Deny”. See Figure 3 for a graphical representation of the explained concepts.

A firewall rule base is a collection of order-sensitive rules. The firewall starts at the top of the rule base until it encounters the first rule that matches the traffic. In a firewall rule, <Source IP>, <Destination IP>, <Destination Port> and <Protocol> can be one value or a range of values. In the remainder of this paper, protocol and port are grouped together in service (for example, TCP port 58 or UDP port 58 are 2 different services).

B. Firewall group objects

Rules containing IP addresses for source/destination and port numbers, are difficult to interpret by humans. Modern firewalls allow the usage of firewall objects, called groups, to give a logical name to a source, a destination, or a port, which is more human-friendly. Groups are populated with IP addresses or ports and can be nested. The groups are used in the definition of the rules. Using groups should improve the manageability of the firewall.

C. Firewall rule relationships

In [6], the following relations are defined between rules:

- **Exactly Matching:** Exactly matching rules (**Rx=Ry**). Rules **Rx** and **Ry** are exactly matched if every field in **Rx** is equal to the corresponding field in **Ry**.
- **Inclusively Matching:** Inclusively matching rules (**Rx⊂Ry**). Rule **Rx** inclusively matches **Ry** if the rules do not exactly match and if every field in **Rx** is a subset or equal to the corresponding field in **Ry**. **Rx** is called the subset match while **Ry** is called the superset match.
- **Correlated**: Correlated rules (**Rx⋈Ry**). Rules **Rx** and **Ry** are correlated if at least one field in **Rx** is a subset or partially intersects with the corresponding field in **Ry**, and at least one field in **Ry** is a superset or partially intersects with the corresponding field in **Rx**, and the rest of the fields are equal. This means that there is an intersection between the address space of the correlated rules although neither one is subset of the other.
- **Disjoint:** Rules **Rx** and **Ry** are completely disjoint if every field in **Rx** is not a subset and not a superset and not equal to the corresponding field in **Ry**. However, rules **Rx** and **Ry** are partially disjoint if there is at least one field in **Rx** that is a subset or a superset or equal to the corresponding field in **Ry**, and there is at least one field in **Rx** that is not a subset and not a superset and not equal to the corresponding field in **Ry**.

 ISBN: 978-1-61208-850-1

Figure 4 represents the differnet relations in a graphical manner. Exactly matching, inclusively matching and correlated rules can result in the following firewall anomalies [8]:

- *Shadowing Anomaly*: A rule **Rx** is shadowed by another rule **Ry** if **Ry** precedes **Rx** in the policy, and **Ry** can match all the packets matched by **Rx**. The result is that **Rx** is never activated.
- *Correlation Anomaly*: Two rules **Rx** and **Ry** can cause a collerlation anomaly if, the rules **Rx** and **Ry** are correlated and if **Rx** and **Ry** have different filtering actions.
- *Redundancy Anomaly*: A redundant rule **Rx** performs the same action on the same packets as another rule **Ry** so that if **Rx** is removed the security policy will not be affected.

A fully consistent rule base should only contain disjoint rules. In that case, the order of the rules in the rule base is of no importance, and the anomalies described above will not occur [6] [7] [8] . However, due to several reasons such as unclear requirements, a faulty change management process, lack of organization, manual interventions, and system complexity [13], the rule base will include correlated, exactly matching, and inclusively matching rules, and thus resulting in evolvability issues.

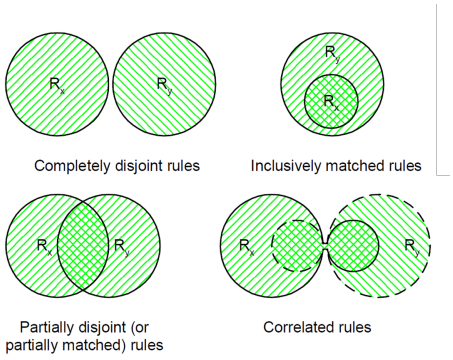


Figure 4. Possible relationships between rules (from [9]).

D. Normalized Systems concepts

Normalized Systems theory [5] [15] originates from the field of software development.

The Normalized Systems Theory takes the concept of system theoretic stability from the domain of classic engineering to determine the necessary conditions a modular structure of a system must adhere to in order for the system to exhibit stability under change. Stability is defined as Bounded Input equals Bounded Output (BIBO). Transferring this concept to software design, one can consider bounded input as a certain amount of functional changes to the software and the bounded output as the number of effective software changes. If the amount of effective software changes is not only proportional to the amount of functional changes but also the size of the existing software system, then NS states that the system exhibits a CE and is considered unstable under change.

Normalized Systems Theory proves that, in order to eliminate CE, the software system must have a certain modular structure, were each module respects four design rules. Those rules are:

- Separation of Concern (SoC): a module should only address one concern or change driver.
- Separation of State (SoS): a state should separate the use of a module by another module during its operation.
- Action Version Transparency (AVT): a module, performing an action should be changeable without impacting modules calling this action.
- Data Version Transparency (DVT): a module performing a certain action on a data structure, should be able to continue doing this action, even is the data structures has undergone change (add/remove attributes).

NS can be used to study evolvability in any system, which can be seen as a modular system and derive design criteria for the evolvability of such a system [16] [17].

III. Requirements for the Solution

This section will discuss the design requirements for an evolution rule base built from the ground up, also known as a green-field approach. These design requirements serve as input for a brown-field approach or convert a non-evolvable rule base into an evolvable rule base.

A. Building an Evolvable Rule Base

In previous work [4], the combinatorics involved when creating a rule base are discussed. For a given network **N**, containing $\mathbf{C_j}$ sources and $\mathbf{H_j}$ destinations, offering 2^{17} services (protocol/port), and having a firewall **F** between the sources and the destinations, it can be shown that $\mathbf{f_{max}}$ is the number of possible rules that can be defined on the firewall **F**:

$$\mathbf{f_{max}} = 2.\left(\sum_{a=1}^{Hj}\binom{Cj}{a}\right).\left(\sum_{a=1}^{Hj}\binom{Hj}{a}\right).\left(\sum_{k=1}^{2^{17}}\binom{2^{17}}{k}\right) \quad (1)$$

where $\mathbf{C_j}$ and $\mathbf{H_j}$ are function of **N**: $\mathbf{C_j} = f_c(\mathbf{N})$ and $\mathbf{H_j} = f_h(\mathbf{N})$

A subset of those rules will respresent the intended security policy and only a subset of that subset will be the set of rules that are disjoint. The maximum size of the disjoint set of "allow" rules (aka a white list) is:

$$\mathbf{f_{disjoint}} = \mathbf{H_j}.2^{17} \quad (2)$$

with $\mathbf{H_j}$ is the number of hosts connected to the network. $\boldsymbol{H_j} = \boldsymbol{fh(N)}$ and 2^{17} the max amount of services available on a host.

The probability that a firewall administrator will always pick rules from the disjoint set is low if there is no conscious design behind the selection of rules.

In previous work [4], based on NS, an artifact is being proposed to create a rule base free of CE for a set of anticipated

changes. The artifact takes the "Zero Trust" [**?**] [**?**] design criteria into account as well, meaning that access is given to the strict minimum: in this case, the combination of host and service.

1) Starting from an empty firewall rule base **F**. Add as first rule the default deny rule **F**[1]= $\mathbf{R}_{\text{default\_deny}}$ with
   - $\mathbf{R}_{\text{default\_deny}}$.Source = ANY,
   - $\mathbf{R}_{\text{default\_deny}}$.Destination=ANY,
   - $\mathbf{R}_{\text{default\_deny}}$.Service= ANY,
   - $\mathbf{R}_{\text{default\_deny}}$.Action = "Deny".
2) For each service offered on the network, create a group. All service groups need to be completely disjoint from each other: the intersection between groups must be empty.
   Naming convention to follow:
   - **S**_*service.name*,
   - with *service.name* as the name of the service.
3) For each host offering the service defined in the previous step, a group must be created containing only one item (being the host offering that specific service).
   Naming convention to follow:
   - **H**_*host.name*_**S**_*service.name*,
   - with *host.name* as the name of the host offering the service
4) For each host offering a service, a client group must be created. That group will contain all clients requiring access to the specific service on the specific host.
   Naming convention to follow:
   - **C_H**_*host.name*_**S**_*service.name*
5) For each **S**_*service.name*,**H**_*host.name*_**S**_*service.name* combination, create a rule **R** with:
   - **R**.Source =**C_H**_*host.name*_**S**_*service.name*
   - **R**.Destination = **H**_*host.name*_**S**_*service.name*
   - **R**.Service= **S**_*service.name*
   - **R**.Action = "Allow"

   Add those rules to the firewall rule base **F**.
   The default rule $\mathbf{R}_{\text{default}}$ should always be at the end of the rule base.

In [4], proof can be found that a rule base created according to the above artifact, results in an evolvable rule base concerning the following set of anticipated changes: addition/removal of a rule, addition/removal of a service, addition/removal of a destination (with or without a new service), the addition of a source. The removal of a source does not impact the rule base but does impact the groups containing the sources.

B. Converting an Existing Rule Base into an Evolvable Rule Base

The previous section describes the green-field situation; builiding a rule base from scratch. The luxery of a green-field is often not present. We require a solution that can convert an existing rule base, into a rule rule base that only contains disjoint rules. Of couse, the original filtering strategy expressed in the rule base must stay the same. From the previous section we know that we require disjoint service definitions. If we can disentangle the service defintions, and adjust the rules accordingly, we have our basic building block for a disjoint rule base. For each disjoint service definition, we need to create as many destination groups as there are host offering that service (lookup in rule base), and for each host-service combination, we require one source group definition. All components are then present to expand a non-evolvable rule base into a normalized evolvale rule base. Figure 5 visualizes what we want the solution to do.

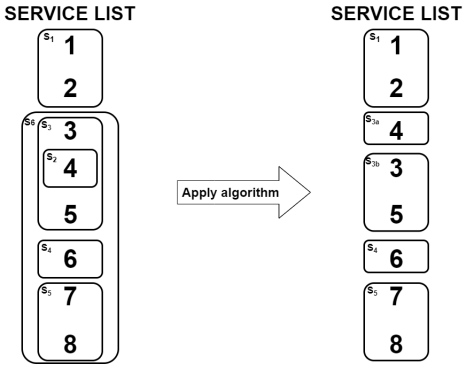


Figure 5. Algorithm objective.

IV. SOLUTION DESIGN

In this section, we will discuss the different components that will make up the algorithm. We start by justifying the choice for Iterated Local Search as metaheuristic [18] [19] [20]. We will discuss the nature of the initial solution, the set of feasible solutions and the objective function associated with a solution. We continue by defining the move type, move strategy, perturbation and stop condition of the Iterated Local Search. The last part of this section discusses the solution encoding and special operations performed in the algorithm, and finally, the presentation of the algorithm.

A. Metaheuristic selection

The objective is to disentangle/reshuffle the service definitions into a set of new service definitions that are disjoint but as large as possible. The simplest solution is to make one service definition per port. But some ports belong together to deliver a service. This logic is somewhere buried in the rule base and service definitions. We may not lose it.

Service definitions containing ports that appear in multiple service definitions must be split in non-overlapping service definitions. The result should be that the degree of overlap (or disjointness) of all service definitions decreases as more service definitions are split. If we measure somehow the degree of disjointness of all the service definitions and see that after a split, the degree of disjointness improved, we know we found a better solution than before.

A Local Search heuristic is a suitable method for organizing such gradual improvement process. To avoid getting stuck in a local optimum (see further), the Local Search will be upgraded to an Iterated Local Search (ILS). The Iteration part should avoid getting stuck in a local optimum where we can no longer perform splits and improve the disjointness. The

iteration part should perform a perturbation, a special kind of split, that will allow the continuation of the search for improvement.

B. Initial Solution and Neighbourhood

The initial solution is given. It is the rule base with all the service definitions. The set of all service definitions is our Neighbourhood. We will have to pick a service definition, check if it is disjoint and if not, split it and see how this affects the solution - disjointness improved or not. The solution space SP for the service definitions consists of all possible combinations of ports. If the number of distrinct ports in the service groups equals P, then the SP is:

$$\mathbf{SP} = \sum_{k=1}^{P} \binom{P}{k} \tag{3}$$

P can be max 2^{17}. We are looking to find a new solution, that is part of the solution space, in which all service definitions are disjoint yet grouped in groups of maximum size.

C. Objective Function

To know if the splitting of a service definition is improving the solution, we need a mechanism to express the degree of disjointness of the service definition. Each definition contains ports and those ports may be part of multiple definitions. We define the port frequency of a port as the number of times this port appears in a service definition. The higher the frequency, the more the need to splitting this port off.

We define the **DI**, the Disnointness Index of a service definition $\mathbf{S_x}$, as the SUM of the port frequencies **PF** of the ports *px* of $\mathbf{S_x}$, divided by the number of ports in $\mathbf{S_x}$.

DI(S_x)=$\frac{\sum_{i=1}^{nx} PF(p_x)}{nx}$ with nx = $\mid S_x \mid$= number of ports in S_x

DI is one if all ports only appear once in a service definition. A DI of one means the service definition is fully disjoint.

We define the Objective Function **OF** as the sum of all **DI** of all service definitions.

OF = $\sum_{i=1}^{n} DI(S_i)$ with n the number of service definitions in the solution.

If the Objective Function value is equal to the number of service definitions, then we have found an optimal solution. Not necessarily a Global Optimum as making service definitions of one port would also yield to an Objective Function equal to the number of service definitions.

D. Feasable Solutions

Whatever kind of splits we will be performing, the original filtering logic of the rule base must be maintained, meaning that splitting service definitions will result in splitting rules to have the identical rule base behaviour. The original rule will have to be removed from the rule base and replaced by a number of rules equal to the splits size (split in 2 groups, 3 groups, etc.).

E. Move Type

The move type will be a split of a service definition. A service definition can:

- be a subset of existing service definitions
- be the superset of existing service definitions
- have a partial overlapping with other service definitions.
- be a combination of the above.

The split during the Local Search will concists of splitting, carving out, all existing subgroups. We call the split the full-carve-out move. This split is chosen as it resolves both the sub and superset case.

Example: A service definition **S**={1,2,3,4,5,6,7}. There also exists service definitions $\mathbf{S_1}$={1,2} and $\mathbf{S_2}$={5,7}. Carving out $\mathbf{S_1}$ and $\mathbf{S_2}$ from **S** gives, $\mathbf{S_1}$= {1,2}, $\mathbf{S_2}$={5,7} and **S'**={3,4,6}

This move type is not able to handle partial overlapping service defintions. It is expected that when all carve outs are done, there will be a number of overlaps remaining that require a diffent kind of move (see furhter).

F. Move Strategy

All services with a **DI** greater than one are candidates for splitting. It seems logical to start splitting the service with the largest **DI**. If that service cannot be split (no subgroups), then the second-largest **DI** is taken, etc. If a group can be split, the impact of the split is calculated. When **OF** improves (=decends), the move is accepted and executed. If not, the next service in the sorted service **DI** list is chosen. The move-strategy is a variant of the First Improvement strategy of the ILS metaheuristic; a variant as we first order the service **DI** list and take the top element from the list.

G. Perturbation

The carve-out of subgroups cannot remove all forms for disjointness. Correlated (partially overlapping) service definitions cannot be split this way. That is why, when no more carve-outs are possible, a new split operator is required. The operator will determine if a service definition overlaps with another service definitions. If it does, the intersection is split-off. By splitting of this intersection, a new service definition will be created that may be inclusively matching with the other service definitions. Another iteration of the Local Search will investigate this and perform the required carve-outs. We consider this kind of split as a perturbation.

H. Stop Conditions

If all possible carve-outs are done, and all perturbations are done, then there are no more inclusively matching and correlated rules. All port frequencies are equal to one, all service group **DI**'s are equal to one, and **OF** will equal the number of service definitions. The solution cannot be improved anymore.

 ISBN: 978-1-61208-850-1

I. Solution Encoding

The algorithm has been implemented in JAVA. The different components of the solution are implemented as JAVA classes. We tried to stay as close as possible to the NS principles by defining data classes, which only contain data and convenience methods to get and set the data, and task classes used to perform actions and calculations on the data objects.

1) Port: Services contain ports. A port is linked to a protocol (TCP or UDP). PortRange is the class representing a range of ports with an associated protocol.

```
public class PortRange
{
private String protocol;
private int begin;
private int end;
}
```

For a single port, begin = end.

2) Port Frequency: Within a solution, each port will have a frequency that is equal to the number of service definitions in which this port appears. PortFrequency is the class representing the port frequency and the service definitions containing that port.

```
public class PortFrequency
{
private int portnumber;
private int frequency;
private ArrayList<String> group_occurancelist;
}
```

3) Port Frequencies list: PortFrequencies class is the list of all ports existing in a solution and for each port the port frequency in the solution. The i^{th} element of the array represents port i. The content of the i^{th} element contains the port frequency information of port i. As there are TCP and UDP ports, two arrays are required for a full port frequency list.

```
public class PortFrequencies
{
private PortFrequency TCP_portfrequency[]=
   new PortFrequency[65536];
private PortFrequency UDP_portfrequency[] =
   new PortFrequency[65536];
}
```

4) Service: The Services class represent a service definition and contains all port ranges, UDP and TCP, associated with the service.

```
public class Service
{
private String name;
private ArrayList<PortRange> udp_ranges;
private ArrayList<PortRange> tcp_ranges;
}
```

5) ServiceList: The ServiceList class is the list of all service definitions of a solution.

```
public class ServiceList
{
private String name;
private ArrayList<Service> servicelistitems;
}
```

6) Service DI: For each service definition, the disjointness index must be calculated and stored. The disjointness index is stored in the Service_DI class.

```
public class Service_DI
{
private Service service;
private double disjointness_index;
}
```

7) ServiceDIList: The ServiceDIList class is a list of all DI's of all service definitions of a solution. This list represents the neighbourhood as this list will be used to iterate over. The service DI list is an ordered list, with the service with the highest DI as the first element of the list.

```
public class Service_DI_List
{
private ArrayList<Service_DI> service_DI_list;
}
```

J. Operations

The algorithm contains a number of tasks that perform actions on and with the data classes. The most important and relevant ones are listed in this section.

1) PortFrequenciesConstructor: The PortFrequenciesConstructore will calculate the port frequencies of all ports used in all services. It takes the current servicelist as input. The result - a PortFrequenciesList - is accessible via a get-method.

2) Service_DI_List_Creator: The Service_DI_List_Creator will calculate the DI of all services. The inputs are the current service list and portfrequencieslist and the result - a ServiceDIList - is accessible via a get-method.

3) Service_Split_Evaluator: The Service_Split_Evaluator will perform a full-carve-out-split. The inputs are the service to split, the current service list, and the current portfrequencieslist. The result of the split, being the a new ServiceList, a new ServiceDIList, a new PortFrequenciesList and the value of the objective function, are accessible via get-methods.

4) Service_Perturbation: The Service_Perturbation will check if a perturbation is possible and if so, perform it. The inputs are the current servicelist and the portfrequencieslist. The results of the split, being the new ServiceList, new ServiceDIList, new PortFrequenciesList and the value of the objective function, are accessible via get-methods.

K. The Iterated Local Search Algorithm

Algorithm 1 (see Figure 6), is the ILS algorihtm designed according to the components described in previous sections. The important variables are:

- sl = the service list.
- pfl = the portfrequencies list.

- sdil = the service DI list.
- of = objective function value of a solution.
- fully_disjoint = boolean indicating if the solution is fully disjoint.
- end_of_neighbourhood = boolean indicating if the full neighbourhood has been seached.
- objective_function_improvement = boolean indicating is the objective function has improved.
- neighbourhoodpointer = index of an element in the sorted neighbourhood
- service_to_split = service of the neighbourhood that will be investigated for splitting.

V. Solution Demonstration

This section starts with describing the platform used to perform the demonstrations, followed by information on the different data sets that are used to run the algorithm and finished with the results of running the algorithm with the three demonstration sets.

A. Demonstration environment

The algorithm is written in JAVA using JAVA SDM 1.8.181, developed in the NetBeans IDE V8.2. The demonstration ran on an MS Surface Pro (5th Gen) Model 1796 i5 - Quad Core @ 2.6 GHz with 8 GB of MEM, running Windows 10.

B. Demonstration sets

1) Demo set: The first data set consists of a manually created list of service definitions. The set contains a lot of exceptions to test the robustness of the algorithm such as services having different names but identical content, services having almost the same name (case differences) but different content, empty service definition, both TCP and UDP ports etc.

2) Engie Tractebel set: Engie Tractebel Engineering delivered the export of a Palo Alto Firewall used in an Azian branch. The set is a realistic representation of a firewall that interconnects a branch office with the rest of the company network.

3) Engie IT data center set: Engie IT delivered the export of a Palo Alto Firewall used in the Belgium Data centre. The firewall connects different client zones in the data centre with a data centre zone containing monitoring and infrastructure management systems.

C. Demonstration results

The demonstration results show the evolution of 3 indicators. The objective function is the main indicator. Also visualized are the level 1 and level 2 iteration indicators. The level 1 indicator is the number of times that the outer DO loop of the algorithm has run. The indicator measures the number of times a perturbation is made. The level 2 indicator is the number of times the inner DO loop of the algorithm runs within a given level 1 iterations. This means that each time a new level 1 iteration runs, the level 2 iterator is reset. The X-axes of the represents the cumulative level 2 iterations.

Algorithm 1: ILS for service list normalization

```
sl = load_initial_solution(filename);
pfl = portfrequen-
  cies_list_constructor(sl).get_portfrequencies_list();
sdil = service_di_list_creator(sl, pfl).get_service_di_list();
of = service_di_list.get_objective_function();
fully_disjoint = FALSE;
end_of_neighbourhood = FALSE;
objective_function_improvement = FALSE;
while NOT full_disjoint AND NOT end_of_neighbourhood
 do
    neighbourhood = sdil.sort;
    neighbourhood_pointer = 1 (top of list)
    objective_function_improvement = FALSE;
    while NOT improvement_objective_function AND NOT
     fully_disjoint AND NOT end_of_neighbourhood do
        servicer_to_split = neighbour-
          hood.get_element(neighbourhood_pointer);
        service_split_evaluator(service_to_split, sdil, pfl);
        objective_function_improvement = ser-
          vice_split_evaluator.get_objective_function_improved();
        if objective_function_improvement = TRUE then
            sl= service_split_evaluator.get_service_list();
            pfl = ser-
              vice_split_evaluator.get_portfrequencies_list();
            sdil =
              service_split_evaluator.get_service_di_list();
            fully_disjoint =
              service_di_list.is_fully_disjoint_check();
        else
            neighbourhood_pointer ++
        end
        end_of_neighbourhood =
          sdil.end_of_list_check(nieghbourhood_pointer);
    end
    if end_of_neighbourhood then
        service_perturbation_exists =
          service_perturbation.perturbation exists(sl,pfl);
        if service_perturbation_exists then
            sl= service_split_evaluator.get_service_list();
            pfl = ser-
              vice_split_evaluator.get_portfrequencies_list();
            sdil =
              service_split_evaluator.get_service_di_list();
            fully_disjoint =
              service_di_list.is_fully_disjoint_check();
            end_of_neighbourhood = FALSE;
        else
            end_of_eighbourhood = TRUE;
        end
    end
end
if fully_disjoint then
    PRINT "Probably the Global Optimum has been
      found";
else
    PRINT "Local Optimum found";
end
PRINT "Solution = " + sl.get_overview();
```

Figure 6. ILS based algorithm

1) Demo set: The algorithm produces its result in 1 to 2 sec. The start value of the objective function is 110, and the end value is 34. The total number of level 1 iterations is 43, and the total number of level 2 iterations is 568. The algorithm starts 28 service definitions. The algorithm ends with 34 service definitions. The objective function goes down in an almost exponential mode. The Level 2 Iterations go up in an almost logarithmic mode, and the Level 2 Iterations follow a kind of saw-tooth function, with a frequency that goes towards the size of the neighbourhood. Figure 7 shows the evolution of the OF, level 1 and level 2 iterations during the exectution of the algorithm.

2) Engie Tractebel set: The algorithm produces its result in 190 sec. The start value of the objective function is 278, and the end value is 62. The total number of level 1 iterations is 23, and the total number of level 2 iterations is 358. The algorithm starts 79 service definitions. The algorithm ends with 62 service definitions. The objective function goes down in staged mode. The Level 2 Iterations go up in an almost logarithmic mode, and the Level 2 Iterations follow a kind of saw-tooth function, with a frequency that goes towards the size of the neighbourhood. Figure 8 shows the evolution of the OF, level 1 and level 2 iterations during the exectution of the algorithm.

3) Engie IT data center set: The algorithm produces its result in 360 sec. The start value of the objective function is 3876 and the end value is 418 . The total number of level 1 iterations is 127 and the total number of level 2 iterations is 10.835. The arlgorithm starts 459 service definitions. The algorithm ends with 418 service definitions. The objective function goes down in staged mode. The Level 2 Iterations go up in an almost logarithmic mode and the Level 2 Iterations follow a kind of saw-tooth function, with a frequency that goes towards the size of the neighbourbood. Figure 9 shows the evolution of the OF, level 1 and level 2 iterations during the exectution of the algorithm.

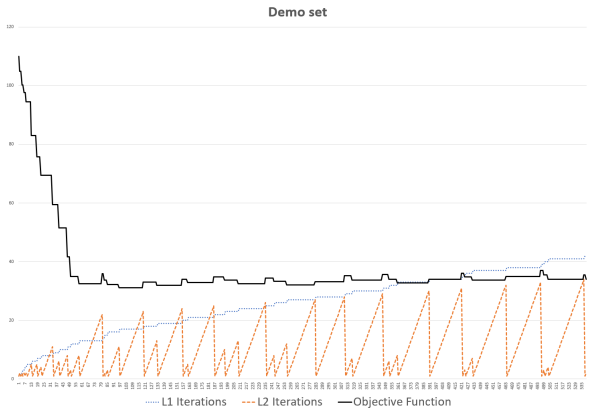


Figure 7. Objective Function, L1 Iteration and L2 Iteration for the demo set.

VI. Solution Evaluation and Discussion

In this section, we will evaluate and discuss the algorithm, starting with the Big O of the algorithm. We continue to specify the impact of the splits on the rule base and by positioning the algorithm as an essential building block in the conversion of a rule base into an evolvable rule base. We conclude by proposing some potential performance enhancement methods and an alternative for the algorithm.

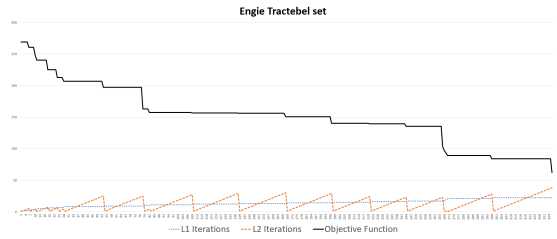


Figure 8. Objective Function, L1 Iteration and L2 Iteration for the Engie Tractebel set.

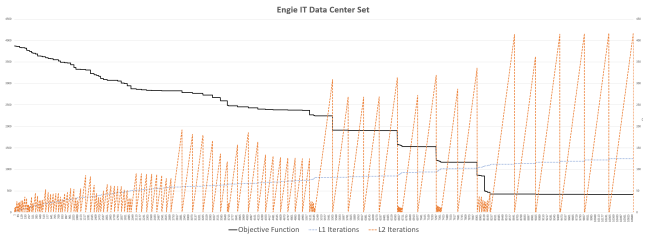


Figure 9. Objective Function, L1 Iteration and L2 Iteration for the Engie IT data center set.

A. Big O of the algorithm

The algorithm contains two nested loops that both can iterate over the full neighbourhood, meaning the algorithm will be quadratic with respect to the size of the neighbourhood. The operations performed in the most inner loop, like Service_DI_list_Creator, Servicer_split_Evalturator are also proportional to the size of the neighbourhood. We can thus conclude that the Big O of the complete algorithm is cubic - $O = n^3$, where n is the size of the neighbourhood (= size of the solution = the number of service definitions)

B. Impact of the splits on the rule base

Each time a service is split, there is an impact on the rule base. All rules containing this service must be adjusted according to the result of the split. Two kinds of adjustments are required.

- Split rules: The rules containing this service must be split in 2 rules that contain the results of the split.
  Example:
  Before split: Rx = source-destination-service
  Split: service splits into service1 and service2
  After split: Rx1 = source-destination-service1 and Rx2 = source-destination-service2

- Rename services: When service splits result in existing services, those services will be renamed to track the changes. All rules that are impacted by this rename must be adjusted. Example:
  Before split: servicex
  Split: service x splits in service x' and service x", but those existed already under the names service xV5 and service yV8. Service xV5 becomes service xV6,

 ISBN: 978-1-61208-850-1

and service yV8 becomes service yV9.
After split: service x is replaced by service xV6 and service yV9. Service xV5 is replaced by service xV6 and service yV8 is replaced by service yV9.

The algorithm does not include the adjustments of the rules, but counts the number of times such an adjustment is required. Table I shows the different test sets, the initial and final value of the objective function, while Table II shows, for the different test sets, the initial rule base size and the number of additional rules due to the splits. Further work is required to adjust the algorithm to perform the actual splits and to have a better view on the actual amount of additional rules.

TABLE I. EVOLUTION OBJECTIVE FUNCTION

Test set	Inital OF	Final OF
Demo set	110	34
Engie Tractebel set	278	62
Engie IT data center set	3874	418

TABLE II. IMPACT ON THE RULE BASE

Test set	Initial size rule base	Extra rules	Service renames
Demo set	NA	74	55
Engie Tractebel set	37	135	152
Engie IT data center set	522	2940	2976

C. Building block for evolvable rule base

The list of disjoint services is the essential building block for building an evolvable rule base. According to artifact of Section III-A, for each service, there should be as many destination groups created as there are hosts offering this service. And for each destination group created in this manner, there should be one source group created. The population of those destination and source group can happen via investigation of the existing rule base.

D. Impact on the size of the rule base

The algorithm has been demonstrated in only 3 test cases. More test cases are required to get a better insight into the impact of splitting services into disjoint servers on the rule base's size. A more detailed study of different firewall types within Engie is on the researcher's agenda.

E. Potential performance improvements

1) Pre-processing: The firewall configuration contains both service definitions and service group definitions. Service groups aggregate service definitions. In the simulations, service groups are part of the service list, and logically those are the first that will undergo the full-carve-out operations. It could be beneficial to exclude service groups. This would require the replacement of the service groups used in the rule base by their individual services and splitting of rules accordingly. This pre-processing step also takes time, and it remains to be seen if it improves performance.

2) Memory: The algorithm would benefit from some memory as defined in metaheuristics. All groups that are disjoint no longer require checking if they are disjoint and can be removed from the search list. This could reduce the size of the neighbourhood dynamically and improve performance.

3) Deterministic approach: Tests of the algorithm show that there is always converges to the same solution for a given initial solution. Although we cannot prove it formally (yet), for a given initial solution, there is convergence to one solution that seems to be the Global Optimum. The creation of the algorithm resulted in a progressive insight about how to disentangle the service definitions. We now believe that the disentanglement can be achieved without the the calculation of the objective function, which is basically saying we no longer have an Iterated Local Search algorithm but an algorithm that will follow a predetermined path toward the solution.

VII. CONCLUSION

Using Iterated Local Search, an algorithm was created that allowed the disentanglement of a set of groups that are nested and overlapping, into a set of groups that is disjoint from each other. Such an algorithm can be applied in the specific context of making firewall rule bases evolvable. The algorithm has been demonstrated successfully. Progressing insight during the creation of the algorithm points toward a deterministic algorithm. More firewall exports are required to get a better idea on the impact of the splitting of services into disjoint services, on the size of the rule base.

REFERENCES

[1] H. Shel and A. Spiliotes, "The State of Network Security: 2017 to 2018", Forrester Research, November 2017

[2] "2018 State of the firewall", Firemon whitepaper, URL https://www.firemon.com/resources/, [retrieved: April, 2021]

[3] "Firewall Management - 5 challenges every company must address", Algosec whitepaper, URL https://www.algosec.com/resources/, [retrieved: April, 2021]

[4] G. Haerens and H. Mannaert, "Investigating the Creation of an Evolvable Firewall Rule Base and Guidance for Network Firewall Architecture, using the Normalized Systems Theory", International Journal on Advances in Security, Volume 13 nr. 1&2, pp. 1-16, 2020

[5] H. Mannaert, J. Verelst and P. De Bruyn, "Normalized Systems Theory: From Foundations for Evolvable Software Toward a General Theory for Evolvable Design", ISBN 978-90-77160-09-1, 2016

[6] E. Al-Shaer and H. Hamed, "Taxonomy of conflicts in network security policies", IEEE Communications Magazine, 44(3), pp. 134-141, March 2006

[7] E. Al-Shaer, H. Hamed, R. Boutaba and M. Hasan, "Conflict classification and analysis of distributed firewall policies", IEEE Journal on Selected Areas in Communications (JSAC), 23(10), pp. 2069-2084, October 2005

[8] M. Abedin, S.Nessa, L. Khan and B. Thuraisingham, "Detection and Resolution of Anomalies in Firewall Policy Rules", Proceedings of the IFIP Annual Conference Data and Applications Security and Privacy, pp. 15–29, 2006

[9] E. Al-Shaer and H. Hamed, "Design and Implementation of firewall policy advisor tools", Technical Report CTI - techrep0801, School of Computer Science Telecommunications and Information Systems, DePaul University, August 2002

[10] S. Hinrichs, "Policy-based management: Bridging the gap", Proceedings of the 15th Annual Computer Security Applications Conference, pp. 209-218, December 1999

 ISBN: 978-1-61208-850-1

[11] A. R. Hevner, S. T. March, J. Park and S. Ram, “Design Science in Information Systems Research”, MIS Quarterly, Volume 38, Issue 1, pp. 75-105, 2004

[12] P. Johannesson and E. Perjons, “An Introduction to Design Science”, ISBN 9783319106311, 2014

[13] W. R. Stevens, “TCP/IP Illustrated - Volume 1 - the Protocols”, Addison-Wesley Publishing Company, ISBN 0-201-63346-9, 1994

[14] H. Zimmermann and J. D. Day, “The OSI reference model”, Proceedings of the IEEE, Volume 71, Issue 12, pp. 1334-1340, Dec 1983

[15] H. Mannaert, J. Verelst and K. Ven, “The transformation of requirements into software primitives: Studying evolvability based on systems theoretic stability”, Science of Computer Programming, Volume 76, Issue 12, pp. 1210-1222, 2011

[16] P. Huysmans, G. Oorts, P. De Bruyn, H. Mannaert and J. Verelst, “Positioning the normalized systems theory in a design theory framework”, Lecture notes in business information processing, ISSN 1865-1348-142, pp. 43-63, 2013

[17] G. Haerens, “Investigating the Applicability of the Normalized Systems Theory on IT Infrastructure Systems, Enterprise and Organizational Modeling and Simulation”, 14th International workshop (EOMAS) 2018, pp. 23-137, June 2018

[18] M. Rafael, P. M. Pardalos and M. G. C. Resende, “Handbook of Heuristics”, ISBN 978-3-319-07123-7 IS, 2018

[19] Z. Michalewicz and D. B. Fogel, “How to Solve It: Modern Heuristics”, ISBN 978-3-642-06134-9, 2004

[20] E. Talbi, “Metaheuristics - From Design to Implementation”, ISBN 978-0-470-27858-1, 2009

 ISBN: 978-1-61208-850-1

Pattern-Based Ontological Transformations for RDF Data using SPARQL

Marek Suchánek
Faculty of Information Technology
Czech Technical University in Prague
Prague, Czech Republic
email: marek.suchanek@fit.cvut.cz

Robert Pergl
Faculty of Information Technology
Czech Technical University in Prague
Prague, Czech Republic
email: robert.pergl@fit.cvut.cz

***Abstract*—RDF data are being described by ontologies (OWL or RDFS) to state the meaning and promote interoperability or so-called machine-readability. However, there are many overlapping ontologies that one can use for a single dataset. To overcome this issue, mappings between ontologies are made to capture the relations forming the overlaps. Such mappings can be used with inference and reasoning tools, but rewriting rules must be applied to transform the dataset. This work proposes a new way of transformations builds on top of the SPARQL query language. It uses defined RDF patterns representing modules that can be interrelated. Our method's primary focus is for larger-scale transformations where existing methods require hard-to-maintain, i.e., non-evolvable mapping definitions. A brief demonstration, as well as a comparison with other transformation languages, is provided.**

***Index Terms*—ontology, transformation, RDF, OWL, versatility, evolvability**

I. Introduction

The technologies around the Resource Description Framework (RDF), including the Web Ontology Language (OWL), are more and more used in various domains and setups. It is a versatile instrument for capturing any knowledge. As RDF is the keystone of Semantic Web and Linked Data, it helps to form interoperable metadata and data. The OWL ontologies and RDF schemas help to describe the internal structure and meaning of related RDF data. Without such descriptions, the RDF data are just a set of triples. On the other hand, with a used ontology, it is possible to make assumptions about types, properties, and even infer new data from existing.

Therefore, more and more ontologies are introduced to different domains, projects, and purposes. Sometimes those ontologies are just in the form of dictionaries, where specific terms are defined for referring to them. In contrast, others are full-fledged ontologies with a definition of properties and special relations, such as subclassing. The overlaps between the ontologies are inevitable just as it needs to evolve over time. Again, the principles of linked data can help with that. Two ontologies can be easily linked together, for example, to define equality of terms just as any other relation.

Issues arise when the underlying RDF data needs to be transformed between such ontologies based on the mapping. The first issue is with the evolvability of the transformation itself, i.e., how to handle changes from the linked ontologies. Then, there are issues with possibilities of executing the transformation over datasets that might contain non-mapped entries. Finally, the consistency between the original and the transformed RDF dataset is also an important issue. There are different approaches to the RDF transformations that we will review and evaluate in this paper, mainly regarding the evolvability.

Section II briefly describes the necessary terminology and existing solutions for transforming RDF data that we use as a source of valuable information and experience. The proposed method for pattern-based ontological transformations if presented in Section III. In Section V, we summarize and discuss the results and outline possible future steps to enhance RDF transformation's evolvability.

II. Related Work

In this section, we briefly introduce important terminology and provide an overview of the current possibilities for RDF data transformations.

A. RDF, RDFS, and OWL

RDF is a set of specifications based on a simple idea that everything can be described using so-called semantic triples or triplets – subject, predicate, object [1]. Such triplets can also be expressed as a graph; therefore, we use also terms as knowledge graph or graph database when talking about RDF data. As it was initially intended as a metadata model, the triples consist of identifier, more specifically Uniform Resource Identifiers (URI), to specific resources and possibly literals on the object position. RDF Schema (RDFS) [2] or the Web Ontology Language (OWL) [3] can be used to define classes, properties, or constraints to bring structure into otherwise unstructured RDF data. Although the OWL capabilities in terms of expressiveness and versatility are much higher than RDFS, both are expressed again as RDF data using triplets. Nowadays, RDF technologies are used widely in many activities and for many kinds of data [2]. RDF, RDFS, and OWL are the basis of Semantic Web and Linked Data but are also used in data-intensive research (e.g., biology or chemistry) or in conceptual modelling for specifying dictionaries and taxonomies [4].

 ISBN: 978-1-61208-850-1

RDF provides versatility in capturing any information both as data and related metadata. For example, there are various well-defined ways of capturing versions of data. Dublin Core [5] defines `hasVersion`, `isVersionOf`, or `valid` terms for annotating the data. There are even more elaborated vocabularies, such as Changeset [6], that allows annotating data with a changelog. However, for OWL ontologies, the version information is usually included both in metadata but also as part of URIs, e.g., `http://purl.org/dc/elements/1.1/date` that is a persistent snapshot, and without version part, the identifier targets the latest, which can change over time. Moreover, OWL provides additional properties for versioning, such as `versionInfo` or `incompatibleWith`.

B. Ontology Mapping

OWL itself defines a fundamental way of mapping. Again, due to the versatility of RDF and the fact that OWL ontologies are captured as RDF data, the mapping can be done using well-defined properties. Using the core principles of RDF, it is possible to import two (or more) ontologies and create a mapping between them or create an extension build on top of them.

For capturing that two individuals have the same identity, `owl:sameAs` can be used. It is essential to mention here that even OWL Classes are themselves individuals, so the property can be used to express class equality on the identity level. For just stating that two classes are equivalent (but not necessarily have the same identity), `owl:equivalentClass` is appropriate. There are many more standard properties to define different relations of (in)equality. As it is still just RDF with given meaning to property, one can define own mapping properties but need to explain the purpose and determine how it should be implemented.

C. STTL: SPARQL-Based Transformation

The approach of [7] is based on the SPARQL Protocol and RDF Query Language. As SQL is well-known and used for querying relational data, SPARQL works similarly (even in terms of syntax) for RDF data, i.e., semantic triples. The SPARQL-Based Transformation (STTL) uses SPARQL extension `TEMPLATE` with additional functions, such as `st:apply-templates`. The transformation allows to query and prepare data for a template using a standard `SELECT` query. Then, it applies a template by the additional functions. Output can be practically any textual format as a template is just a string with marks where place specific data as shown in Figure 1. The authors present examples, including RDF-to-HTML or RDF-to-RDF/Turtle.

Nevertheless, for just RDF-to-RDF transformations, the standard `CONSTRUCT` query can be used. It avoids installing the extension, yet it provides a simple way of a mapping definition. The RDF Data are prepared using the `WHERE` clause and then used to build new triples. One can see such a SPARQL query also as a text; therefore, it can be synthesized using another tool or script and even executed programmatically.

```
TEMPLATE {
    "The triple is: " ?x ", " ?p ", " ?y "!"
}
WHERE {
    ?x ?p ?y
}
ORDER BY ?x ?p ?y
```

Fig. 1. Example of STTL template query

D. RML: RDF Mapping Language

The RDF Mapping Language (RML) [8] is a generic mapping language, based on and extending R2RML. Although it is still in state of unofficial draft, it captures interesting and novel ideas related to RDF mapping. The main focus is on transforming data from textual formats, such as CSV, XML, or JSON into RDF. The mappings in RML are captured as RDF triples as shown in Figure 2. First, there is a definition of a data source, such as CSV file. Then, there are several mappings for subjects, predicates, and objects that capture what classes and properties are used and how the data are queried from the data source [9] [10].

```
<#PersonHobbyMapping>
  rml:logicalSource [
    rml:source "http://ex.com/people.json";
    rml:referenceFormulation ql:JSONPath;
    rml:iterator "$.person.hobby.[*]"
  ];

  rr:subjectMap [
    rr:template "http://ex.com/{Hobby_ID}";
    rr:class ex:Hobby
  ].
```

Fig. 2. Example of RML mapping for JSON source

E. TRIPLE Language

A Query, Inference, and Transformation Language for the Semantic Web (TRIPLE) [11] is an older representative of RDF transformation language. In contrast to the previous two approaches, TRIPLE has its foundations in mathematics and logics. It can be expressed both through RDF (e.g. RDF/XML) and mathematical syntax as it is an extension of Horn logic similar to F-Logic. The models (i.e. sets of triples) are first class citizens in TRIPLE. It has several useful functions for defining the transformations, such as reification of statements or path expressions. Unfortunately the tooling support is not easily available but it has been also used for business agents [12].

III. TRANSFORMING RDF DATA USING PATTERNS AND SPARQL

In this section, we describe our design of the pattern-based ontological transformations for RDF data using SPARQL as

well as a prototype implementation. The advantages of our approach when compared to using plain SPARQL or other solutions are discussed in Section V. First, we outline set the problem statement and related requirements. Then, steps of our method are explained in the subsequent subsections.

A. The Problem and Requirements

Our work aims to design a transformation method that takes a set of triples as input together, and using a set of transformation rules produces a new set of triples. Although SPARQL `CONSTRUCT` query allows defining such rules, it does not provide re-usability of definitions nor modularization. Our solutions must promote evolvability by design as a fine-grained modular structure concerning the Normalized Systems Theory [13], especially the Separation of Concerns and Data Version Transparency principles. However, SPARQL already provides a fair way of querying RDF data that should be re-used for defining the transformation patterns as visualized in Figure 3.

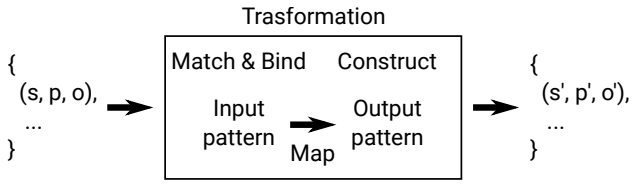


Fig. 3. Architecture of the pattern-based RDF transformation

B. Patterns Mapping Definition

In our method, each transformation pattern is formed by two sets of triples – input and output. The principle and syntax are the same as in SPARQL; the input pattern can bind variables in any position (subject, predicate, object). Then, the output triples can contain those variables multiple times. RDF data are traversed, and the input pattern is filled, i.e., variables are substituted by values. When the whole input pattern is matched, the output pattern is populated with the values for variables and RDF data are added to the result dataset. For defining both input and output patterns, standard RDF prefix definitions might be necessary (but separately for each). So far, the operation is the same as with a set of SPARQL `CONSTRUCT` queries.

To introduce a fine-grained modular structure from trivial cases, each pattern has separated input and output parts, as shown in Figure 4 and Figure 5. The variable names from the input part are exposed, and the output part can use them. It must be defined what input is used (cases with multiple input parts are explained further). Using variables in this way can be seen as a coupling inside the module.

C. Pattern Modules and Submodules

The definition of input pattern must take into account that RDF/OWL works with open-world assumptions. For example,

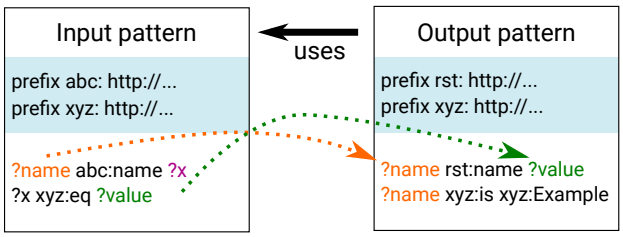


Fig. 4. Concept of relating input and output patterns

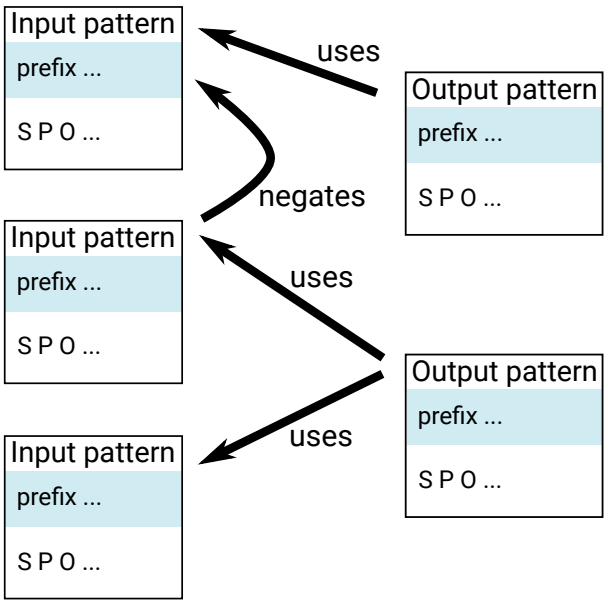


Fig. 5. Combining input patterns for output patterns

if the second pattern should be used only if the first one is not matched, it must include negating the first input triples. To do this efficiently, the input triples must be shareable across patterns. By importing triples, one must comply with its defined variable names or rename them. In a similar way, it can serve to separate the concerns when the same input produces multiple different outputs for a single dataset, e.g., a name of a person is turned into the name of the personal folder at HR and the name on a cup in a shared kitchen.

We call a set of related patterns in this way a *patterns module* and each of its part is a submodule. In a trivial case without any shared part, a pattern forms itself (input and output triples with prefix definitions) a module. As input can re-use multiple other inputs (with renaming the variables), the output can also define various input definitions to be used, as shown in Figure 5.

 ISBN: 978-1-61208-850-1

D. Generating and Executing SPARQL CONSTRUCT

The pattern definitions and modules capture the knowledge of what inputs should be used for producing what outputs. With that, it is possible to generate a set of SPARQL `CONSTRUCT` queries to be executed in arbitrary order over an input RDF dataset. There are several steps to this procedure for each patterns module:

1) Resolve imports in all input definitions, including variable renaming.
2) For each output definition, import input definition(s) including variable renaming.
3) Merge used prefixes and use renaming mechanism for conflicts (when name and URI do not match).
4) Generate SPARQL CONSTRUCT query with input part in WHERE clause.
5) Execute the query over the input dataset and add result into output dataset.

E. Prototype Implementation

A prototype of the method has been implemented in Python using rdflib [14] and Jinja2 [15] templating language as a CLI application. It takes a folder with pattern modules and an RDF source (a file or SPARQL endpoint) and produces an output RDF file. The project folder has a subfolders per each module, where files are prefixed by `input_` and `output_`. The re-usability of definitions is realized through variables and imports in Jinja2. Each of the modules forms a single Jinja2 environment where each output file is loaded with substituted Jinja2 variables. The compiled transformation pattern is then turned using an internal template into a SPARQL query and executed over a graph imported from the input file or using the given SPARQL endpoint.

The patterns are executed actually in alphabetical order as that is the way the filesystem is traversed. We added a possibility to randomize the order of transformation to demonstrate that it is irrelevant. After all of the patterns are successfully applied, the output file is written out. Several issues might appear when running the transformation. For instance, the pattern's definition may contain a syntactic error; in that case, the application reports where the problem is. For the SPARQL endpoint, there might be connection issues. Finally, there might also be a problem when output definition is using a variable that is not defined in any of the inputs.

IV. Example Case: FOAF and vCard

To demonstrate the use of our method and prototype, we want to transform a dataset about people created using Friend-of-a-Friend ontology FOAF [16] into a dataset according to the vCard ontology [17]. Some of the presented patterns are intentionally not optimal in terms of complexity to show the use of additional features. The examples put together related patterns forming a single module, as explained previously. Individual patterns are marked using comments starting with # symbol.

A. Simple Personal Information

The first example in Figure 6 shows a trivial case where is just a single input pattern and a single output pattern. For every person with a name from the input dataset, a name according to vCard is created. The used syntax is simplified RDF Turtle with custom directives. Here `@input` serves to specify the only input pattern based on its name. No variable renaming is used. During the transformation, this case is simply turned into a SPARQL query, i.e., prefixes are merged together (no conflicts), the input pattern is inserted into `WHERE` clause, and the output becomes the body of `CONSTRUCT`.

```
# Input pattern: input_simple_foaf
@prefix rdf:
 <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .

?person rdf:type foaf:Person .
?person foaf:name ?name .

# Output pattern: output_simple_vcard
@input: input_simple_foaf .
@prefix vcard:
 <http://www.w3.org/2006/vcard/ns#> .

?person vcard:fn ?name .
```

Fig. 6. Trivial case for single input pattern to single output pattern

B. Negated Input Pattern

Figure 7 demonstrates a case where the second input pattern negates the first one using `@negate` directive. The SPARQL query takes the first input pattern and wraps it using `NOT EXISTS` construct. This case also has a conflict in prefixes as the second input pattern is using a different FOAF version. During the processing, the conflicting prefixes are renamed before turned into a SPARQL query.

```
# Input pattern: input_foaf_person_and_org
@prefix rdf:
 <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .

?agent rdf:type foaf:Person .
?agent rdf:type foaf:Organization .

# Input pattern: input_foaf_membership
@negate: input_foaf_person_and_org .
@prefix foaf: <http://xmlns.com/foaf/0.2/> .

?agent foaf:member ?member .

# Output pattern: output_simple_vcard
@input: input_foaf_membership .
@prefix vcard:
 <http://www.w3.org/2006/vcard/ns#> .

?agent vcard:member ?member .
```

Fig. 7. Example of using a negated input

C. Multiple Input Patterns

The final example Figure 8 shows multiple input patterns used for an output pattern. It creates both person triples and organization triples (with names). The second `@input` directive is enhanced with a variable renaming dictionary to avoid name clashes. In this case, all variables are renamed, but it is not mandatory, and only some can be renamed. Both input patterns use the same FOAF prefix, and thus it will be merged without the need of solving conflicts.

```
# Input pattern: input_foaf_person
@prefix rdf:
 <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .

?person rdf:type foaf:Person .
?person foaf:name ?name .

# Input pattern: input_foaf_organization
@prefix rdf:
 <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .

?organization rdf:type foaf:Organization .
?organization foaf:name ?name .

# Output pattern: output_complex
@input: input_foaf_person .
@input: input_foaf_organization
        {organization: org, name: orgName} .
@prefix rdf:
 <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix vcard:
 <http://www.w3.org/2006/vcard/ns#> .

?person rdf:type vcard:Individual .
?person vcard:fn ?name.

?org rdf:type vcard:Organization .
?org vcard:title ?orgName.
```

Fig. 8. Example with multiple inputs and renaming variables

V. Discussion

This section summarizes the advantages of our approach and confronts it with the other existing solutions introduced in Section II. The planned future steps are outline as well.

A. Evolvability and Maintainability

Our solution is designed to allow the definition of interlinked patterns for transformations. Each pattern relates an input set of triples where some variables are bound, and the output set of triples is used. If the patterns are not related to each other, there are no ripple effects internally. Also, thanks to the declarative approach of RDF, the order of triples and, thus, an order of transformation execution is irrelevant. To mitigate ripple effects of relating the patterns through sharing (or negating) input triples, we introduced pattern modules where if an input set is changed, it requires a change in submodules but not other modules. To allow transformation with intermediate layer of temporary triples, our solution can be used in a pipeline, e.g., first, transform input to some intermediary triples and then transform intermediary triples to output. The order of such transformations is, of course, necessary, but internally the order of executing a transformation pattern is still irrelevant, i.e., there are no dependencies.

As for the practical maintainability of the transformation defined as a set of patterns, the key is the overall solution's simplicity. Each pattern can be defined as a standalone file with input and output or re-use other definition on input. When there is an update of some of the imports (e.g. a target ontology changes), it spreads over patterns as a cross-cutting concern. However, only patterns related to the target ontology are required to be updated, which cannot be avoided. Versioning of a set of patterns (a project) can be done easily through standard VCS tools, including branching or version tagging.

B. Comparison to Existing Solutions

When compared to the solutions mentioned in Section II, our approach has a different focus. RML [8] targets transforming any data to RDF. To use it for our purpose, we would need to define a mechanism for using RDF as a data source and navigating through it, which would be additional overhead. Moreover, mappings' definitions do not provide such maintainability as in our case, despite other similarities. STTL [7] allows us to query RDF data and transform it into any textual format, which would again cause additional overhead here as we would need to synthesize RDF. This functionality is supported directly by SPARQL `CONSTRUCT`.

Mappings on the OWL-level, such as `owl:sameAs`, can be used together with our method as is shown in Section IV. It provides a generalized way of executing transformations based on those mappings (and others defined in transformation patterns). Finally, TRIPLE [11] has a concept of models as the building block, whereas we similarly use patterns. Our solution can be in mathematical terminology seen as a set of functions (each realizing a pattern transformation). It takes a set of triples and gives a set of triples. On the contrary to this last method, we also implemented the prototype actually to execute the transformations.

C. Future Steps

The presented approach, together with the prototype implementation, can be already used for executing the transformations, as shown in Section IV. However, we plan to enhance the ease of writing the transformation patterns as well as executing the transformation over various kinds of data sources (e.g. local files or SPARQL endpoints). The first of the use cases that will be taking advantage of our transformation solution is working with different conceptual models encoded in RDF. The ultimate goal here is to develop mappings between metamodels so the models can be integrated and transformed on the semantical level. As we expect non-trivial sets of transformations patterns required to achieve the goal, it will also serve as a verification and a good source of potential enhancements to our method. Finally, we plan to continue

 ISBN: 978-1-61208-850-1

in this effort to create a user-friendly editor for specifying, testing, and executing the pattern transformations.

VI. Conclusion

This paper proposed and demonstrated a new approach for transforming RDF data using patterns specified for SPARQL `CONSTRUCT` queries. It was shown that our solution's evolution and maintainability is improved as the core principles of Normalized Systems were taken into account. When compared to other existing methods for transforming RDF data, the primary focus is slightly different. Whereas other targets transform other data formats into RDF or vice versa, our contribution is supporting the transformation of RDF data between different ontologies where basic mapping predicated from OWL is not sufficient. Moreover, our approach's advantages are the most visible when used with more complex use cases where many and overlapping transformation patterns are needed. Finally, we plan to continue in this effort to create a user-friendly editor for specifying and testing the pattern transformations.

Acknowledgment

The research was supported by Czech Technical University in Prague grant No. SGS20/209/OHK3/3T/18.

References

[1] A. Hogan, *The Web of Data*. Springer International Publishing, 2020.

[2] Dan Brickley and Ramanathan V Guha and Brian McBride, "Rdf schema 1.1," *W3C recommendation*, vol. 25, pp. 2004–2014, 2014.

[3] P. Hitzler et al., "OWL 2 Web Ontology Language Primer," *W3C recommendation*, vol. 27, no. 1, p. 123, 2009.

[4] S. Powers, *Practical RDF*. O'Reilly Media, Incorporated, 2003.

[5] Dublin Core Metadata Initiative, "Dublin Core™ Metadata Element Set, Version 1.1: Reference Description," 2012. [Online]. Available: https://www.dublincore.org/specifications/dublin-core/dces/ 2021.03.30

[6] S. Tunnicliffe and I. Davis, "Changeset," 2009. [Online]. Available: https://vocab.org/changeset/ 2021.03.31

[7] O. Corby and C. Faron-Zucker, "STTL - A sparql-based transformation language for RDF," in *WEBIST 2015 - Proceedings of the 11th International Conference on Web Information Systems and Technologies, Lisbon, Portugal, 20-22 May, 2015*, V. Monfort, K. Krempels, T. A. Majchrzak, and Z. Turk, Eds. SciTePress, 2015, pp. 466–476. [Online]. Available: https://doi.org/10.5220/0005450604660476 2021.03.29

[8] A. Dimou, M. V. Sande, P. Colpaert, R. Verborgh, E. Mannens, , and R. V. de Walle, "RML: A generic language for integrated RDF mappings of heterogeneous data," in *Proceedings of the Workshop on Linked Data on the Web co-located with the 23rd International World Wide Web Conference (WWW 2014), Seoul, Korea, April 8, 2014*, ser. CEUR Workshop Proceedings, vol. 1184. CEUR-WS.org, 2014. [Online]. Available: http://ceur-ws.org/Vol-1184/ldow2014_paper_01.pdf 2021.03.25

[9] B. D. Meester, P. Heyvaert, R. Verborgh, and A. Dimou, "Mapping languages: Analysis of comparative characteristics," in *Joint Proceedings of the 1st International Workshop on Knowledge Graph Building and 1st International Workshop on Large Scale RDF Analytics co-located with 16th Extended Semantic Web Conference (ESWC 2019), Portorož, Slovenia, June 3, 2019*, ser. CEUR Workshop Proceedings, vol. 2489. CEUR-WS.org, 2019, pp. 37–45. [Online]. Available: http://ceur-ws.org/Vol-2489/paper4.pdf 2021.03.22

[10] P. Heyvaert, A. Dimou, R. Verborgh, E. Mannens, and R. V. de Walle, "Towards approaches for generating RDF mapping definitions," in *Proceedings of the ISWC 2015 Posters & Demonstrations Track co-located with the 14th International Semantic Web Conference (ISWC-2015), Bethlehem, PA, USA, October 11, 2015*, ser. CEUR Workshop Proceedings, vol. 1486. CEUR-WS.org, 2015. [Online]. Available: http://ceur-ws.org/Vol-1486/paper_70.pdf 2021.03.22

[11] S. Decker et al., "TRIPLE - an RDF rule language with context and use cases," in *W3C Workshop on Rule Languages for Interoperability, 27-28 April 2005, Washington, DC, USA*. W3C, 2005. [Online]. Available: http://www.w3.org/2004/12/rules-ws/paper/98 2021.03.28

[12] M. Sintek and S. Decker, "Using TRIPLE for business agents on the semantic web," *Electronic Commerce Research and Applications*, vol. 2, no. 4, pp. 315–322, 2003. [Online]. Available: https://doi.org/10.1016/S1567-4223(03)00040-1

[13] H. Mannaert, J. Verelst, and P. D. Bruyn, *Normalized Systems Theory: From Foundations for Evolvable Software Toward a General Theory for Evolvable Design*. Kermt (Belgium): Koppa, 2016.

[14] RDFLib Team, "rdflib 5.0.0," 2020. [Online]. Available: https://rdflib.readthedocs.io 2021.03.31

[15] Pallets, "Jinja Documentation (2.11.x)," 2021. [Online]. Available: https://jinja.palletsprojects.com 2021.03.31

[16] D. Brickley and L. Miller, "FOAF Vocabulary Specification 0.99," 2014. [Online]. Available: http://xmlns.com/foaf/spec/ 2021.03.31

[17] R. Iannella and J. McKinney, "vCard Ontology - for describing People and Organizations," 2014. [Online]. Available: https://www.w3.org/TR/vcard-rdf/ 2021.03.31

 ISBN: 978-1-61208-850-1

Exploring the Use of Code Generation Patterns for the Creation of Evolvable Documents and Runtime Artifacts

Herwig Mannaert and Gilles Oorts
Normalized Systems Institute
University of Antwerp, Belgium
Email: herwig.mannaert@uantwerp.be

Koen De Cock and Peter Uhnak
Research and Development
NSX BVBA, Belgium
Email: koen.de.cock@nsx.normalizedsystems.org

***Abstract*—Many organizations are often required to produce large amounts of documents in various versions and variants. Though many solutions for document management and creation exist, the streamlined automatic generation of modular and evolvable documents remains challenging. It has been argued in previous work that a meta-circular metaprogramming architecture enables a modular creation of source artifacts with very limited programming. In this contribution, a proof of concept is explored to generate modular LaTeX documents from runtime information systems through the use of a reduced version of this metaprogramming environment. The actual generation of several basic administrative document sources is explained, and it is argued that this architecture can easily be applied to generate other types of source artifacts using live runtime data.**

***Index Terms*—Evolvability; Normalized Systems Theory; Metaprogramming; Document Creation; Single Sourcing**

I. Introduction

Organizations are often required to produce large amounts of versions and variants of certain documents. While they have traditionally focused their efforts into streamlining technical product documentation [1], they are now also looking to build business value by creating personalized customer-faced documents [2]. At the same time, current information systems are producing massive amounts of relatively simple documents, e.g., invoices and timesheets, based on corporate data.

The streamlined and possibly automatic generation of such documents leads to concepts like modularization and single sourcing [1], both reminiscent of similar techniques used in the creation of software. Just like in software, dealing with versions and variants requires the design of document structures that deplete the rippling of changes in order to provide a level of evolvability [3]. Moreover, the use of parameter data during the instantiation of document variants seems similar to the inner workings of code generation environments.

In our previous work, we have presented a meta-circular implementation of a metaprogramming environment [4], and have argued that this architecture enables a scalable collaboration between various metaprogramming projects featuring different meta-models [5][6]. In this contribution, we investigate the use of a reduced version of this code generation environment within the generated software applications. More specifically, we explore the creation of evolvable artifacts, such as documents, where runtime data of the generated software application is used to instantiate the artifacts.

The remainder of this paper is structured as follows. In Section II, we briefly discuss some aspects and terminology related to the creation and single sourcing of documents, an important class of artifacts created by information systems at runtime. In Section III-A, we explain the basic concept of Normalized Systems Theory with regard to the design of evolvable artifacts. Section III-B recapitulates the architecture of our meta-circular code generation environment, and explains that this expansion of source code artifacts is not limited to programming code. Section IV presents how the generation environment can be configured to instantiate and expand runtime artifacts such as documents using live data. Finally, we present some conclusions in Section V.

II. Modular and Evolvable Document Creation

While organizations have traditionally focused their efforts in document management into streamlining product documentation [1], there is a widespread belief that personalized customer-faced documents can build business value by enhancing customer loyalty [2]. However, repurposing internal documents to be used for online purposes such as sales, marketing, product documentation and customer support has proven to be diffcult. Moreover, it is hard to find any best practices or repeatable models developed that address this challenge [1]. In this section, we briefly discuss some techniques and issues regarding the creation of evolvable documents.

A. Document Creation and Single Sourcing

A successful approach to handle any complex system or problem is modularization [7][8]. An example of such an approach in the area of document management is *Component Content Management (CCM)*, defined as *a set of methodologies, processes, and technologies that rely on the principles of reuse, granularity, and structure to allow writers to author, review, and repurpose organizational content as small components* [1]. One of the fundamental ideas of component content management is the separation of content and layout [9]. The granularity of a component in CCM is defined by the smallest unit of usable information [10]. Several standards exist that

 ISBN: 978-1-61208-850-1

define practical and technical implementation guidelines for creating modular and reusable content. According to Andersen and Batova [1], the most widely implemented standard is the *Darwin Information Typing Architecture (DITA)*.

Originally regarded as the broader discipline of CCM in the early 2000s, *single sourcing* has been defined as one of the fundamental aspects of CCM concerned with the design and production of modular, structured content. An elaborate description of single sourcing and its concepts, advantages, methodology, guidelines and practical examples, can be found in [11]. There are three fundamental aspects to single sourcing. First, content is made *reusable* by separating content from format. A second aspect is *modular writing*. Content is written in stand-alone modules instead of whole documents. This allows content to be assembled into documents from *singular source files that contain unique content*, the third aspect of single sourcing. Besides *assembling* the content modules into documents, i.e., combining source files in a hierarchical and sequential way with a distinct combination of audience, purpose and format, the modules need to be *linked*, i.e., connected to make them into coherent documents.

Enabling the content creators to focus on the actual substance of documents instead of having to deal with layout and publishing technologies, should lead to various advantages: saving time and money, improving document usability, and increasing team synergy [11]. Single sourcing recognizes two types of document creation. *Repurposing* entails merely reusing content modules for a different output format. *Re-assembly* on the other hand, is a more impactful way of reusing modules to develop documents for different purposes or audiences. Contrary to repurposing, re-assembly also includes changing the sequence of modules, the conditional inclusion, and the hierarchical level of inclusion.

B. Modular and Parametrized Document Generation

The emergence of concepts like modularization, CCM, and single sourcing regarding the management of certain classes of documents, e.g., technical documentation or personalized documents, is highly reminiscent of similar concepts in software codebases. Indeed, software developers have been striving for decades to modularize codebases, to separate concerns into singular source files, and to assemble source code modules into software applications, in a continuous effort — or quest — to reuse and repurpose these source modules.

Both documents and software source bases can have successive *versions* in time that contain additions, corrections or omissions to its content, and can be branched into concurrent *variants* when variations in content and/or purpose occur. Just like in software, dealing with versions and variants of a document requires the design of document structures to provide a desired level of evolvability. Evolvable documents are documents that do not hinder or limit the application of changes made to their structure or content. They are free from ripple effects that would cause changes to the documents to be highly difficult and costly [3].

Documents, such as technical and/or personalized documents, can have many concurrent variants. In technical documents, these variants range from the variation or even conditional presence of entire technical descriptions and procedures due to differences in the components of various installations, to simple parameter values like the serial number, color, or location of the documented installation or product. But short and simple documents, like letters, invoices or timesheets, can also be considered to have many variants due to different parameter values. This aspect of parameter-based or *model-based instantiation of document variants*, is highly reminiscent of environments for *code generation in software*.

III. Expansion of Evolvable Modular Structures

In this section, we discuss the expansion and assembly of evolvable modular structures. We introduce *Normalized Systems Theory (NST)* as a theoretical basis to design information systems —and conceptually other kinds of modular structures— with higher levels of evolvability, and its realization in a framework to generate and assemble programming code, and possibly other types of source artifacts.

A. Normalized Systems Theory and Evolvable Structures

NST was proposed to provide an ex-ante proven approach to build evolvable software [12], [13], [14]. It is theoretically founded on the concept of *systems theoretic stability*, a well-known systems property demanding that a bounded input should result in a bounded output. In the context of information systems, this implies that a bounded set of changes should only result in a bounded impact to the software. This implies that the impact of changes to an information system should only depend on the size of the changes to be performed, and not on the size of the system to which they are applied. Changes causing an impact dependent on the size of the system are called *combinatorial effects*, and considered to be a major factor limiting the evolvability of information systems. The theory prescribes a set of theorems and formally proves that any violation of any of the following *theorems* will result in combinatorial effects (thereby hampering evolvability) [12], [13], [14]:

- *Separation of Concerns*
- *Action Version Transparency*
- *Data Version Transparency*
- *Separation of States*

Applying the theorems in practice results in very fine-grained modular structures in software applications, which are in general difficult to achieve by manual programming. Therefore, the theory also proposes a set of patterns to generate significant parts of software systems which comply with these theorems. More specifically, NST proposes five *elements* that serve as design patterns for information systems [13][14]:

- *data element*
- *action element*
- *workflow element*

 ISBN: 978-1-61208-850-1

- *connector element*
- *trigger element*

Based on these elements, NST software is generated in a relatively straightforward way. Due to this simple and deterministic nature of the code generation mechanism, i.e., instantiating parametrized copies, it is referred to as *NS expansion* and the generators creating the individual coding artifacts are called *NS expanders*. This generated code can be complemented with custom code or *craftings* at well specified places (anchors) within the skeletons or boiler plate code. This results in the structural separation of four dimensions of variability [14][6]:

1) *Mirrors* representing data and flow models, using standard techniques like ERD (Entity Relationship Diagram) and BPMN (Business Process Model and Notation).
2) *Skeletons* expanded by instantiating the parametrized templates of the various element patterns.
3) *Utilities* corresponding to the various technology frameworks that take care of the cross-cutting concerns.
4) *Craftings* or custom code to add non-standard functionality that is not provided by the skeletons.

It has been extensively argued that the design theorems and structures of NST are applicable to all hierarchical modular architectures that exhibit cross-cutting concerns [15]. More specifically related to documents, the software theorems and element patterns of NST are very similar to the principles of CCM, that rely on reuse and fine-grained modular structures to allow writers to author, review, and repurpose organizational content as small components, and to the concept of single sourcing, demanding the separation of content and layout. Moreover, it has been shown that the application of NST to the design of evolvable document management systems leads to architectures that are in accordance with the principles of CCM and single sourcing [3], [16], [17].

B. Meta-Circular Code Generation or Artifact Expansion

NST has been realized in software through a code generation environment to instantiate instances of the various elements or design patterns. Due to the simple and deterministic nature of this code generation, i.e., instantiating parametrized copies, it is referred to as *NS expansion*. We have also argued that metaprogramming or code generation environments exhibit a rather similar and straightforward internal structure [5], [6], distinguishing:

- *model files* containing the model parameters.
- *reader classes* to read the model parameter files.
- *model classes* to represent the model parameters.
- *control classes* to select and invoke the generator classes.
- *generator classes* instantiating the source templates, and feeding the model parameters to the source templates.
- *source templates* containing the parametrized code.

As the NST metaprogramming environment was developed for the creation of web information systems, it has always included the generation of various building blocks, e.g., reader and model classes, which are similar to those of the code generation environment itself. This has made it possible to merge those generated code modules with the corresponding code generation modules, thereby evolving the metaprogramming environment into a meta-circular architecture [4]. This meta-circular architecture, described in [4], [5], [6], is schematically represented in Figure 1 and entails several advantages. First,

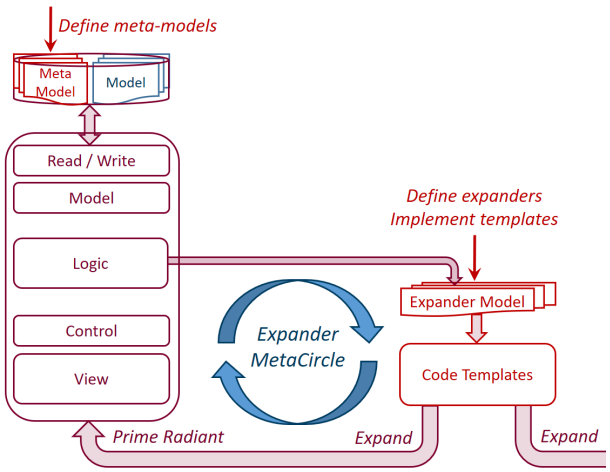


Fig. 1. The meta-circular architecture for NS expanders and meta-application.

this architecture enables the *regeneration of the metaprogramming code itself*, thereby avoiding the growing burden of maintaining the often complex meta-code, such as adapting it to new technologies. Second, it allows for a structural decoupling between the two sides of the code generation transformation, i.e., the domain models and the code generating templates. This also removes the need for contributors to get acquainted with the — basically non-existing —internal code structure of the metaprogramming environment, as additional expanders with corresponding coding templates can be defined and activated using a declarative control mechanism.

Moreover, the definition of additional meta-models and/or templates is not limited to programming code either. Instead of containing *Java* or *JavaScript* code, the templates may just as well correspond to hierarchical document modules such as chapters and sections, containing commands and settings of typesetting systems like Markdown/Pandoc or LaTeX. And any model representing parameter data and/or small content components may serve as a meta-model and drive the expansion or instantiation of the document. The meta-circular architecture does not require any explicit programming to support the new model entities representing the document. As we have seen, the various classes corresponding to the new model entities (XML readers and writers, model classes, control and generator classes) will be automatically generated.

IV. Exploring Runtime Expansion of Artifacts

In this section, we explore the assembly or expansion of parametrized documents using the NST meta-circular code generation environment.

 ISBN: 978-1-61208-850-1

A. Document Creation and Information Systems

As explained in Section II, an interesting duality exists between information systems and document creation. Information systems often support the creation of simple documents, such as invoices or timesheets, incorporating data that is entered and managed within the information system. At the same time, the streamlined creation of large amounts of document variants, for instance in the case of technical product documentation, requires some tooling to specify and manage the various parameters driving the creation of the document variants. In other words, information systems often create documents, and document creation systems usually require a supporting information system.

For the exploratory development targeted at the creation of documents using the NST meta-circular code generation environment, we have opted for the first scenario. The streamlined creation of variants of complex documents would require the definition of an elaborate meta-model describing the structure and domain parameters of the documents. The creation of such a model is out of scope of this contribution, but does not seem to pose a significant risk. Therefore, we decided to explore the generation of rather simple documents based on common data entities like invoices or timesheets. Nevertheless, this explorative development does address a possible and important technological hurdle. As these documents need to incorporate runtime data from the live information systems, e.g., the actual details of the various invoices, this proof of concept validates the expansion of artifacts based on runtime data from any information system expanded by the NST metaprogramming environment. In this way, the development can also serve as a validation for the expansion of other source artifacts based on live runtime data of information systems, such as marketing emails or sensor configuration files.

B. Declarative Control and Runtime Expansion

Consider two samples of a simplified data model for an administrative information system as presented in Figure 2.

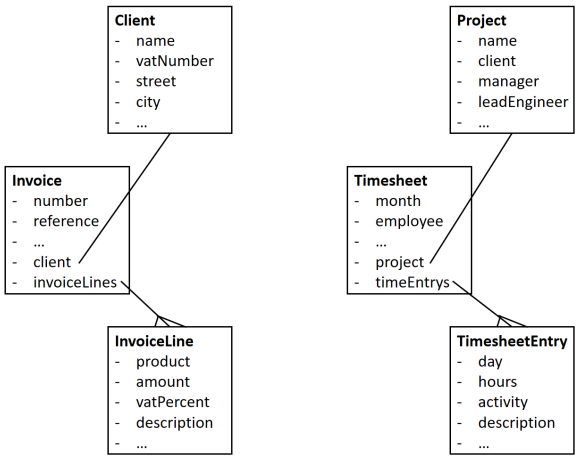


Fig. 2. Samples of a simplified data model for an administrative system.

- An *invoice* with some attributes, e.g., an invoice number and reference, containing a reference to a client, and consisting of several invoice lines.
- A *timesheet* with some attributes, e.g., the month and employee, containing a reference to a project, and consisting of several timesheet entries.

These data entities are expanded into *data elements*, collections of software classes as described in [6], by the NST metaprogramming environment, and incorporated in an information system. The expanded data elements or collections of classes include:

- Reader and writer classes to read and write the XML data files, e.g., *InvoiceXmlReader* and *InvoiceXmlWriter*, *TimesheetXmlReader* and *TimesheetXmlWriter*.
- Model classes to represent and transfer the various entities, and to make them available as an object graph, e.g., *InvoiceDetails* and *InvoiceComposite*, *TimesheetDetails* and *TimesheetComposite*.
- View and control classes to perform *CRUDS (create, retrieve, update, delete, search)* operations in a generated table-based user interface.

In the same way that the instances of the NST meta-model data elements are read and made available as an object graph at the time of code generation, the instances of the data elements represented in Figure 2 can be made available as an object graph at runtime in a generated information system. Incorporating the core templating engine of the NST metaprogramming environment [6] allows to evaluate the various attributes of the administrative data entities using *Object-Graph Navigation Language (OGNL)* expressions, and to feed them to the (LaTeX) templates that are used to create the invoice and timesheet documents.

As explained in [6], the expansion of artifacts, e.g., source code or document files, is based on a generic *ArtifactExpander* that uses declarative control to evaluate the model parameters and insert them into the source templates. Every individual expander generating a source artifact is defined in an *Expander* XML document. An example of the definition of such an individual expander to expand a LaTeX source file for an invoice is shown below. It is quite similar to the declaration of an expander creating a Java source file during code generation, but has a *TEX* source type and uses for instance the runtime invoice number to construct the filename.

```
<expander name="TexInvoiceExpander"
  xmlns="http://normalizedsystems.org/expander">
  <packageName>net.palver.latex.invoice</packageName>
  <layerType name="ROOT"/>
  <technology name="COMMON"/>
  <sourceType name="TEX"/>
  <elementTypeName>Invoice</elementTypeName>
  <artifact>Invoice-$invoice.number$.tex</artifact>
  <artifactPath>$expansion.directory$/
                $artifactSubFolders$</artifactPath>
  <isApplicable>true</isApplicable>
  <active value="true"/>
</expander>
```

The evaluation of the various instance parameters or attributes is based on OGNL expressions and defined in a separate *ExpanderMapping* XML document. This ensures the separation of content from format, as required by [9] to have reusable and evolvable documents. An example of the definition of such an individual mapping document for the invoice creation is shown below. Besides simple OGNL expressions, it allows to evaluate logical expressions, e.g., whether the invoice client is foreign for VAT purposes, and to make lists of linked objects and their attributes available, e.g., invoice lines.

```
<mapping
 xmlns="https://schemas.normalizedsystems.org/
                xsd/expanders/2021/0/0/mapping">
 <value name="info" eval="invoice.info"/>
 <value name="number" eval="invoice.number"/>
 <value name="client" eval="invoice.client.name"/>
 <value name="vatNr" eval="invoice.client.vatNr"/>
 <value name="street" eval="invoice.client.street"/>
 <value name="city" eval="invoice.client.city"/>
 <value name="isForeign"
  eval="!invoice.client.country.equals('Belgium')"/>
 <list name="invoiceLines"
       eval="invoice.invoiceLines"
       param="invoiceLine">
  <value name="info" eval="invoiceLine.info"/>
  <value name="product" eval="invoiceLine.product"/>
  <value name="amount" eval="invoiceLine.amount"/>
 </list>
</mapping>
```

The values as defined in the expander mapping document are passed to the LaTeX templates. As described in [5], the NST environment uses the *StringTemplate (ST)* engine library. This library supports the creation of a modular document structure by providing *subtemplate include* statements, enabling the document designers to adhere to the principles of single sourcing [11]. For instance, we share the declaration of various LaTeX packages and the definition of some basic commands through the use of the subtemplates `<basePackages()>` and `<baseCommands()>`. And the various invoice lines of an invoice (or timesheet entries of a timesheet) are created by instantiating a corresponding subtemplate for every list item through `<invoiceLines:invoiceTableLine()>` (or `<timesheetEntrys:timesheetTableLine()>`).

A reduced version of the NST metaprogramming environment was integrated into a runtime installation of an expanded information system that included the data elements represented in Figure 2. Based on live data from this runtime environment, tens of LaTeX sources for invoices and timesheets were successfully generated through the use of the expander declarations and parameter evaluations as presented above. It is clear that this expansion architecture allows information systems to create other type of source artifacts based on live runtime data. Indeed, as the NST expansion environment is agnostic with respect to the source type, e.g., able to create LaTeX source documents in exactly the same way as Java source files, the generation of other types of source modules is basically reduced to creating other types of templates. As possible use cases, we mention the creation of HTML emails for marketing purposes, and the assembly of configuration files that can be uploaded to remote IoT sensors or controllers.

V. Conclusion

Many organizations are often required to produce large amounts of versions and variants of documents in areas like technical documentation and accreditation. At the same time, corporate information systems are producing massive amounts of relatively simple documents based on corporate data. The streamlined and possibly automatic generation of such documents leads to concepts like modularization and single sourcing, which are similar to techniques used in code generation software. As in software, dealing with versions and variants requires the design of document structures to deplete the rippling of changes in order to provide a desired level of evolvability.

In our previous work, we have presented a meta-circular implementation of a metaprogramming environment, and have argued that this architecture can be used for code generation based on different and even newly defined meta-models. In this contribution, we have investigated the use of this code generation environment within the generated information systems at runtime. More specifically, we have explored the creation of evolvable artifacts, such as simple administrative documents, where runtime data of the generated software application is used to instantiate the artifacts.

We have shown in this contribution how a reduced version of the NST metaprogramming environment can be integrated within the runtime environment of an expanded information system, and how object graphs containing runtime data can be passed to source templates. We have demonstrated that we can use this runtime metaprogramming environment to successfully produce sources for administrative documents through declarative definitions and OGNL evaluations, without requiring dedicated software programming.

This paper provides different contributions. First, we validate that it is possible to use the NST metaprogramming environment to create another type of source code artifacts, e.g., LaTeX documents. Moreover, we have explained that this implementation adheres to several fundamental concepts regarding modular and evolvable document creation, like CCM and single sourcing. Second, we show that we can integrate a reduced version of the NST metaprogramming environment into a runtime information system expanded by the NST metaprogramming environment, and to generate source artifacts from live data within this running information system.

Next to these contributions, it is clear that this paper is also subject to a number of limitations. We have only demonstrated a single case of generating another type of source artifacts, i.e., LaTeX documents, outside the scope of the generation of software programming code. Moreover, the generated documents are quite simple, and in line with documents that are currently generated by mainstream information systems. Nevertheless, this explorative proof of concept can be seen as an architectural pathfinder, and we are planning to extend both the scope and

 ISBN: 978-1-61208-850-1

size of the generated documents, and the range of possible source types of artifacts that are generated.

REFERENCES

[1] R. Andersen and T. Batova, "The current state of component content management: An integrative literature review," IEEE Transactions on Professional Communication, vol. 58, no. 3, 2015, pp. 247–270.

[2] S. Abel and R. A. Bailie, The Language of Content Strategy. Laguna Hills, CA, USA: XML Press, 2014.

[3] G. Oorts, Design of modular structures for evolvable and versatile document management based on normalized systems theory. Antwerp, Belgium: University of Antwerp, 2019.

[4] H. Mannaert, K. De Cock, and P. Uhnák, "On the realization of meta-circular code generation: The case of the normalized systems expanders," in Proceedings of the Fourteenth International Conference on Software Engineering Advances (ICSEA), November 2019, pp. 171–176.

[5] H. Mannaert, C. McGroarty, K. De Cock, and S. Gallant, "Integrating two metaprogramming environments : an explorative case study," in Proceedings of the Fifteenth International Conference on Software Engineering Advances (ICSEA), October 2020, pp. 166–172.

[6] H. Mannaert, K. De Cock, P. Uhnák, and J. Verelst, "On the realization of meta-circular code generation and two-sided collaborative metaprogramming," International journal on advances in software, vol. 13, no. 3-4, 2020, pp. 149–159.

[7] H. Simon, The Sciences of the Artificial. MIT Press, 1996.

[8] C. Y. Baldwin and K. B. Clark, Design Rules: The Power of Modularity. Cambridge, MA, USA: MIT Press, 2000.

[9] D. Clark, "Content management and the separation of presentation and content," Technical Communication Quarterly, vol. 17, no. 1, 2007, pp. 35–60.

[10] F. Sapienza, "A rhetorical approach to single-sourcing via intertextuality," Technical Communication Quarterly, vol. 16, no. 1, 2007, pp. 83–101.

[11] K. Ament, Single Sourcing: Building Modular Documentation. Norwich, NY, USA: William Andrew Publishing, 2003.

[12] H. Mannaert, J. Verelst, and K. Ven, "The transformation of requirements into software primitives: Studying evolvability based on systems theoretic stability," Science of Computer Programming, vol. 76, no. 12, 2011, pp. 1210–1222, special Issue on Software Evolution, Adaptability and Variability.

[13] ——, "Towards evolvable software architectures based on systems theoretic stability," Software: Practice and Experience, vol. 42, no. 1, 2012, pp. 89–116.

[14] H. Mannaert, J. Verelst, and P. De Bruyn, Normalized Systems Theory: From Foundations for Evolvable Software Toward a General Theory for Evolvable Design. Koppa, 2016.

[15] H. Mannaert, P. De Bruyn, and J. Verelst, "On the interconnection of cross-cutting concerns within hierarchical modular architectures," IEEE Transactions on Engineering Management, 2020, pp. 1–16.

[16] G. Oorts, H. Mannaert, and P. De Bruyn, "Exploring design aspects of modular and evolvable document management," in Proceedings of the Seventh Enterprise Engineering Working Conference (EEWC), May 2017, pp. 126–140.

[17] G. Oorts, H. Mannaert, and I. Franquet, "Toward evolvable document management for study programs based on modular aggregation patterns," in Proceedings of the Ninth International Conferences on Pervasive Patterns and Applications (PATTERNS), February 2017, pp. 34–39.

 ISBN: 978-1-61208-850-1

Trust Patterns In Modern Web-API-based Service Architectures - More Than Technical Security Aspects

Sandro Hartenstein
Berlin School Of Economics And Law
Berlin, Germany
email:
sandro.hartenstein@hwr-berlin.de

Steven Schmidt
DB Station&Service AG / Berlin School Of Economics And Law
Berlin, Germany
email:
s_schmidts19@stud.hwr-berlin.de

Andreas Schmietendorf
Berlin School Of Economics And Law
Berlin, Germany
email:
andreas.schmietendorf@hwr-berlin.de

***Abstract*— This idea paper describes the current perception of Security Patterns and the authors view on the need, to broaden this technical view to a more wholesome approach – Trust Patterns, *integrating* security features. Explaining the need for this approach, it is shown, how this would influence the user's perception of security features through trustworthiness aims towards a consumed service.**

Keywords-Trustworthiness; Digitalization; Information Systems; Security Management; Society.

I. INTRODUCTION

This idea paper aims on emphasizing the importance of trust and trustworthiness of systems in contrast of solely secure designs in a functional way. Security Patterns should be taken into consideration when utilizing Trust Patterns, but a broader view beyond technological aspects into socio-technical or even sociological sides of security and correlating trustworthiness enables a richer and sustained effect and impact towards the user.

Hill and O´Conner define trust in their journal article *A Cognitive Theory of Trust* as follows:

"Trust by definition entails a willingness by the [trustor] to make herself vulnerable to the possibility that another will act to her detriment" [1, p. 28]

A large part of the rapid digitization of services is enabled by the use of WebAPIs. By orchestrating partial services into a full application via apis, it is possible to reduce the effort required compared to a full implementation of all aspects. With this setting, the WebAPIs must be trustworthy in order to be successful. The digitization depends on the well-being of the users. So trustworthy apis are needed, especially due to the rising complexity and in transparency of current and emerging digital services. Trust towards a WebAPIs generates a higher likeliness of using the WebAPI regularly, repeatedly and by recommendation, which are all factors aside from classical security aspects. To relate to the previous quote: Improving trustworthiness not only by security measures but a broader and whole view, increases the chances of consumption and usage.

This paper has the following structure. In the second section, the initial definitions and views are given. In the third section, the required preliminary work is explained. In the fourth section the concept is presented. In the fifth section, the main facts are briefly explained and a planned research project on this topic is presented.

II. TERMS AND VIEWPOINTS

A pattern is an idea that has proven itself in one practice and is likely to be useful to others. There are security design patterns that address typical security challenges and there are trust patterns that address typical trust antecedents.

A pattern typically addresses the process, product, and/or resources. For example, there are security patterns for encrypted transport of data. The communication over this encrypted connection to the user is not part of the pattern. However, the user needs this information to build up trust and to recognize the value subjectively. Therefore, from our point of view, the trust pattern for encrypted transport of data consists of the technical security part and the communicative promotion part in consequence.

Further security pattern deals with the authorization of a webapi access, like the token-based OAuth approach. Another aspect is related to a federated identity management like the application of SAML (Security Assertion Markup Language) or the implementation of a single sign on approach with OpenID and Keycloak.

A trust pattern can exist without a technical part. For example, the reputation of the service provider strongly influences the trust towards his services [2].

The difficulty of creating trust patterns is, that the impact of trust solutions is difficult to prove. The effects that create trust are far more complex than, for example, security, and thus harder to measure. Plus, trust building measures have not always been implemented explicitly, if even. Due to this, a record of past implementations and their possible successful impact will be hard to determine.

Patterns typically addresses the process, product, and/or resources. Trust patterns also address all dimensions and should cover trust in a holistic way.

III. RELATED WORK

Patterns characterised by [3, p. 3] as follows: A pattern is both a spatial configuration of elements that solve a particular problem and a set of associated instructions to create that configuration of elements as effectively as possible. Patterns represent proven and optimal solutions to given problems. This assumes

 ISBN: 978-1-61208-850-1

that these solutions and concepts have been successfully applied again and again in the past.

In order for patterns to become successful or resilient, they must be evaluated [4, p. 4].

For this purpose, the Patterns are evaluates after each step in the lifecycle, defined by [5]. The lifecycle begins with the theory and the specific domain knowledge from which a pattern is developed. This is then deployed and applied. The experiences from the application in use are used in the development [5].

In the development phase, the evaluation is carried out with expert review. In the deployment phase, evaluation takes place with a workshop and peer review. In the operational phase, experiments and surveys are used to check the requirements for patterns [4, p. 5].

Hoffmann's research team published twenty Trust Patterns in 2012.These patterns are templates for defined requirements. For example, a trust pattern, named data usage, is the provision of information on how data is used by the system for the recommendation. Another trust pattern is, for example, the self-explanatory button icon, which states that a button correctly describes the further behaviour of the system. The trust pattern, named setting options, is the provision of personal settings to customise the system [6, pp. 8-10].

These patterns are based on the influencing factors, called Antecedents of Trust, of Söllner et. al, Lee and See and Muir. In relation to the Trust Patterns examples, Understanding, Predictability and Personalisation for the user are the respective arguments [7] [8] [9].

23 principles and 47 patterns for trusted user interfaces has been compiled and prepared in 2018 [10]. The interactive online repository contains not only the content of the patterns, but also meta-information about their origin and links to other patterns and principles [11]. It is also available via webapi, so that it can be easily integrated into development environments. A good example is *Warn When Unsafe*. This pattern addresses informing the user when the configuration of the system is unsafe. It provides for the user to be informed periodically. The frequency of the warnings is very important so that the user notices it but does not get used to it. This is implemented by monitoring the configuration and a safe reference value. Linked patterns are *Attactive Options, Immediate Notifications, Conveying Threats & Consequences, General Notifications About Security, Immediate Options* and *Separating Content*. This pattern originated in Garfinkel's PhD thesis [12] [13].

In summary, it can be said that trust patterns are already being developed and applied in some areas, such as marketing and user interface, due to economic interests. Also, the technical security aspects are also mostly already researched and published. From our point of view, a holistic approach to WebAPIs is missing, which is necessary for the establishment of trustworthy WebAPIs.

IV. CONCEPTUAL CONSIDERATIONS

Our conceptual reasoning is that a holistic, multidimensional view of the trustworthiness of WebAPIs can add great value to digitization. Patterns provide a good way to address non-functional and functional requirements for developing, marketing, and communicating WebAPIs.

Following the trust aspects of software, shown in Figure 1, the product related trust patterns should address the WebAPI relevant ones. Applied security mechanisms such as OAuth 2 address the attributes confidentiality and non-repudiation. The composability is characterized by patterns that reveal the degree of coupling and possibly also the dependencies at the interface. Other features relate to the data processing of WebApis' downstream algorithms, for example, a verification pattern may require the transmission of hash values, thus promoting data integrity.

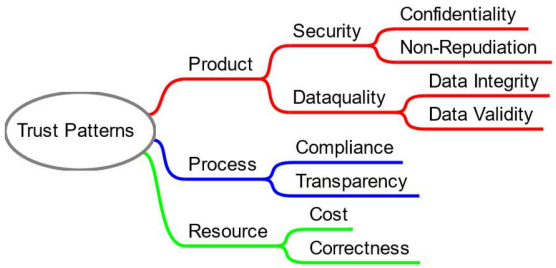


Figure 1 Classification of with the trustworthiness attributes from [14, p. 547].

Process-related trust patterns are intended to the way WebAPIs and downstream services are created. For example, patterns of information sharing about the development process, as well as vulnerability management, are purposeful. Requirements for process certification and specification can also have a confidence-building effect and should be offered as patterns. Another example is patterns for disclosing roadmaps of WebAPIs so that developers can prepare for possible API deactivation or serious changes.

Resource related Trust Pattern do not directly address the WebAPIs, but rather the environment. For example, the corporate brand is an important indicator of trustworthiness and thus of trust. The usage behavior, the number of users, as well as the user types of a WebAPIs is also relevant for trust and should be enhanced with appropriate patterns. The scalability of a WebAPI will also contribute to its distribution and usage, thus promoting trust, should be addressed with patterns.

In a next step, the idea of building a catalog of trust patterns is pursued by carrying out an empirical study. This study is divided into several areas. On the one hand, the most important WebAPIs available are to be examined and, on the other hand, a survey of consumers (typically software developers) of WebAPIs is to provide information about the priorities of the choice of use.

V. CONCLUSION AND FUTURE WORK

Testing trust in services is an important part of creating and establishing trust patterns. For this reason, trust-building measures should always be linked to an evaluation.

Müller and his research colleagues conducted a study on the impact of decentralized blockchain technology on trust in collaboration. This technological view confirms the connection between technology, understanding and trust. In this regard, further trust-oriented technologies should be investigated [15]. Assessments such as these motivate the action, to

 ISBN: 978-1-61208-850-1

perform own examinations of the role of trust in various fields, which have typically only been conducted under security viewpoints in the past. To form a structured approach towards such examinations and findings, the Berlin School of Economics and Law founded a research project determined to an "*empirical evaluation of a model of trustworthiness*" (*orig.: Empirische Untersuchungen zur Modellierung von Vertrauenswürdigkeit*) – EUMoVe [16].

REFERENCES

[1] C. A. Hill and E. A. O'Hara O'Connor, "A Cognitive Theory of Trust," *SSRN Journal*, 2005, doi: 10.2139/ssrn.869423.

[2] S. Schmidt, "Creating a trustworthy public WLAN - approach and partial results [orig.: Schaffung eines vertrauenswürdigen, öffentlichen WLANs - Herangehensweise und Teilergebnisse]," in *Berliner Schriften zu modernen Integrationsarchitekturen*, vol. 24, *ESAPI 2020: 4. Workshop Evaluation of Service-APIs*, A. Schmietendorf and K. Nadobny, Eds., 1st ed., Düren: Shaker, 2020, pp. 35–48.

[3] M. Schumacher, *Security patterns: Integrating security and systems engineering*. Chichester, England, Hoboken, NJ: John Wiley & Sons, 2006, ISBN: 978-0-470-85884-4. [Online]. Available: http://site.ebrary.com/lib/alltitles/docDetail.action?docID=10300660.

[4] A. Hoffmann, H. Hoffmann, and M. Söllner, "Fostering Initial Trust in Applications: Developing and Evaluating Requirement Patterns for Application Websites," *21st European Conference on Information Systems (ECIS), Utrecht, The Netherlands*, vol. 2013. [Online]. Available: https://www.alexandria.unisg.ch/228935/1/Hoffmann%20et%20al.%202013.pdf.

[5] S. Petter, D. Khazanchi, and J. D. Murphy, "A design science based evaluation framework for patterns," *SIGMIS Database*, vol. 41, no. 3, pp. 9–26, 2010, doi: 10.1145/1851175.1851177.

[6] A. Hoffmann, H. Hoffmann, and M. Söllner, "TWENTY SOFTWARE REQUIREMENTPATTERNS TO SPECIFY RECOMMENDERSYSTEMS THAT USERS WILL TRUST," *ECIS 2012 Proceedings.Paper 1*, 2012.

[7] M. Söllner and J. M. Leimeister, *15 years of measurement model misspecification in trust research? A theory based approach to solve this problem.* [Online]. Available: http://pubs.wi-kassel.de/wp-content/uploads/2013/03/JML_189.pdf [retrieved: 01, 2020].

[8] J. D. Lee and K. A. See, "Trust in automation: designing for appropriate reliance," *Human factors*, vol. 46, no. 1, pp. 50–80, 2004, doi: 10.1518/hfes.46.1.50_30392.

[9] B. M. MUIR, "Trust in automation: Part I. Theoretical issues in the study of trust and human intervention in automated systems," *Ergonomics*, vol. 37, no. 11, pp. 1905–1922, 1994, doi: 10.1080/00140139408964957.

[10] L. Lo lacono, M. Smith, E. von Zezschwitz, P. L. Gorski, and P. Nehren, "Consolidating Principles and Patterns for Human-centred Usable Security Research and Development," in *Proceedings 3rd European Workshop on Usable Security*, London, England, Apr. 2018.

[11] L. Lo lacono, *USecured Tools.* [Online]. Available: https://das.th-koeln.de/usecured [retrieved: 01, 2021].

[12] S. L. Garfinkel, *Design Principles and Patterns for Computer Systems that are Simultaneously Secure and Usable*, 2005, ISBN: . Accessed: Feb. 1 2021. [Online]. Available: https://simson.net/thesis/thesis.pdf.

[13] L. Lo lacono, *Warn When Unsafe Pattern.* [Online]. Available: https://das.h-brs.de/usecured/patterns/warn-when-unsafe [retrieved: 02, 2021].

[14] N. Gol Mohammadi *et al.,* "An Analysis of Software Quality Attributes and Their Contribution to Trustworthiness," *Proceedings of the 3rd International Conference on Cloud Computing and Services, Science*, pp. 542–552, 2013.

[15] M. Müller, N. Ostern, and M. Rosemann, "Silver Bullet for All Trust Issues? Blockchain-Based Trust Patterns for Collaborative Business Processes," in *Lecture Notes in Business Information Processing, BUSINESS PROCESS MANAGEMENT: Blockchain and robotic process automation*, A. Asatiani et al., Eds., [S.l.]: SPRINGER, 2020, pp. 3–18.

[16] S. Hartenstein, S. Schmidt, and A. Schmietendorf, "Towards an Empirical Analysis of Trustworthiness Attributes in the Context of Digitalization," in *ICDS 2020 : The Fourteenth International Conference on Digital Society*, pp. 112–116.

 ISBN: 978-1-61208-850-1

A New Algorithm Which Runs in Linear Time Enables the Transformation of Legacy Equipment Into Autonomous and Trustworthy IoTs

Ole Kristian Ekseth
Department of Computer Science (IDI)
NTNU
Trondheim, Norway
oekseth@gmail.com

Erik Morset
Winns
Trondheim, Norway
Erik@winns.no

Svein-Olaf Hvasshovd
Department of Computer Science (IDI)
NTNU
Trondheim, Norway
sophus@ntnu.no

***Abstract*—-The time-cost of today's classification algorithms are all too high: the use of existing algorithms makes it impossible for cloud-based systems to provide decision-support for remote sensors. Thus, there is a need to develop new algorithms with sufficient accuracy, and with explainable outcomes. Thereby, enabling improved utilization of industrial/physical equipment through smart control. In this work we address this requirement: this work presents a new methodology for learning, and training, of classification algorithms. The results indicate that the algorithm outperforms existing methods by 10,000x. Importantly, the new algorithm has a memory footprint considerably smaller than similar strategies, and is straightforward to validate for trustworthiness. This makes it possible to deploy the algorithm at both IoTs and in the cloud, thereby ensuring its broad applicability.**

***Index Terms*—Approximate Computing, performance, execution-time, signal and image processing, segmentation and clustering, machine learning, algorithms, correlation, similarity-metrics.**

I. Introduction

Today, there is an increasing focus on autonomous regulation of sensors: in the energy sector, there is a direct link between automated regulation versus the heating bill [1]. However, the shift from systems with a high degree of manual maintenance to automated sensor logic makes the systems vulnerable to penetration attacks [2]. A recent lapse in penetration security led to "the compromise of 1.9 billion records" [3]. Examples of penetration attacks are:

1) malicious firmware upgrades parameters, *e.g.*, to make use of vulnerabilities in the remote device management interface [4], [5];
2) reading of sensitive sensor data [5], [6];
3) manipulation of actuators through compromising raw or processed sensor data [7].

This requires accurate, fast, and trustworthy algorithms. The problem is that existing algorithms for AI can not be used to control many of today's industrial facilities. This is due to the limited processing power of industrial equipment, combined with issues in data bandwidth, and challenges in certifying algorithms for AI. This paper seeks to address this issue through the design of a new model for classification algorithms.

The increased focus on AI has spurred approaches for automated event detection and prevention [8]. The global sensor market is expected to reach $287.00 Billion by 2025 [9]. Suppliers of industrial control systems are subjected to the same technical challenges, as seen for issues in low data throughput [10] and computational cost of algorithms [11], [12]. Hence, addressing issues in data analysis is bound to significantly increase the value of companies addressing this challenge.

This argues for the design of algorithms applicable to legacy *Internet / Intelligence of Things* (IoTs): if we transform existing equipment into autonomous control units (*e.g.*, to control the heating of hospitals), the result is a reduced amount of traffic on low-latency networks, hence reducing the impact of malicious hacking. By proving existing sensor-components with the flexibility of configurations, one reduces the frequency of firmware updates. Through the use of explainable AI, equipment owners (*e.g.*, owners of real estate) get trust in equipment, thus, enabling the certification (and application) of the systems to environments requiring a high degree of uptime. Therefore, if one manages to design a classification algorithm based on these criteria, the result is an increased accuracy of sensors, *i.e.*, without introducing threats to cybersecurity.

To address these requirements, this paper explores a new methodology for construction of AI. The scope of this paper is as follows: Section II outlines the contributions of this paper, Section III evaluates existing strategies for tuning legacy equipment into smart IoTs, Section IV describes a new $O(n)$ algorithm for tuning dumb equipment into smart IoTs (enabling a 10,000x+ reduction in execution-time), Section V evaluates the accuracy and applicability of the guidelines, Section VI relates the findings to requirements of autonomous IoTs, while Section VII summarizes the findings.

II. Contributions by this paper

The paper presents a new algorithm for the learning and training of classification algorithms. This work exemplifies

 ISBN: 978-1-61208-850-1

how to apply Approximate Computation without loss in prediction accuracy. The paper describes how the system may be applied to industrial systems. The paper identifies a strategy for Approximate Computing that is generalized for a wider audience. The method seeks to intersect algorithms blind spots with knowledge of usage patterns and the physical properties of the data. From the results, we observe how the result is a framework applicable to devices with low computational power, such as IoT networks for control of energy systems. The paper presents results focused on:

1) Approximate Computing: identify generic strategies to simplify the calculation-steps in algorithms;
2) execution-time: identify an algorithm which correctly classifies data in $O(n)$ time (for a data-set with n points);
3) accuracy and provenance: explore the new $O(n)$ algorithm through the classification of images, signals, and generalized application to the MNIST data.

In the following sections, the above perspectives are outlined. The paper relates the concepts of algorithms shortest-paths, combined with Approximate Computing, and knowledge of algorithms Pareto Boundary, to identify a strategy applicable to legacy IoTs, hence enabling existing systems (*e.g.*, sensors controlling heating-systems) to become autonomous.

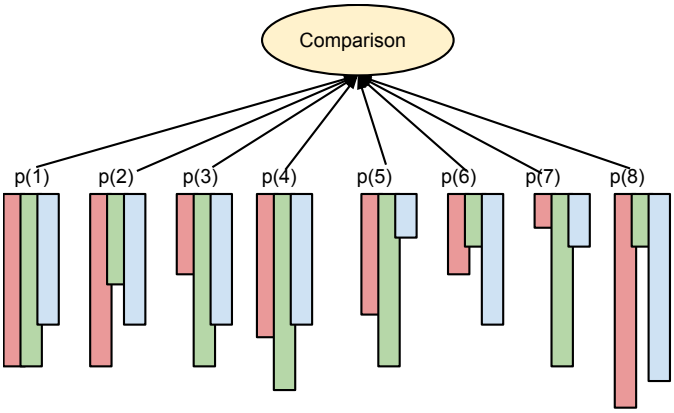


Fig. 1. Why the classification strategy is fast and trustworthy. This figure exemplifies domain-simplifications which makes it possible to reduce the cost of algorithms.

III. Related Work

This paper seeks to identify a strategy for turning legacy equipment into autonomous units. The motivation is to enable autonomous decision-support of infrastructure-critical equipment, for which legacy equipment needs to be updated with new functionality. To reduce the cost of infrastructure (*e.g.*, the heating-bill) there is a need to turn dumb equipment into smart systems, which requires situational awareness of how systems are used [13]. This implies a shift from an analysis of isolated components (*e.g.*, actuators, the pressure of refrigerants, etc.), and into a bird's-eye understanding; if the issues in the performance of classification-algorithms are not resolved, then this task is impossible.

A. Classification Algorithms

There exist numerous clustering-algorithms to use for data-classification:

1) clustering and categorization: a classification algorithm needs to relate clusters of points to an organization reflecting a particular shape (*e.g.*, the shape of letters, the traits of a particular cancer type, etc.);
2) generic algorithms versus domain-application: algorithms are designed towards generic use-cases (*e.g.*, to randomly subdivide points into 'k' number of clusters, as applied in K-means), which results in high error-rate (of generic applications for specialized use-cases);
3) accurate predictions: the specificity of particular use-cases (*e.g.*, interpretation of large versus small feature-differences) makes it important to tailor generic classification-algorithms towards each domain-specific application.

For applications tailored towards IoT and Big Data, the classification algorithm needs to run fast, while having sufficient accuracy. The importance of this requirement is found in how the computers are tightly glued to the physical equipment. A use-case is seen for Winns (a producer of energy systems): Winns reports that the utilization of heating equipment is tightly related to the situational awareness [1].

B. Strategies for Cyber Security in Industrial Systems

When automating infrastructure-critical systems (*e.g.*, heating of hospitals), we need to ensure that the system's behavior (under different circumstances) is correct. Otherwise, the systems can fail spectacularly, as seen at the height of the cold war in 1982 [14]. This involves paying attention to:

1) Penetration Security and System Integrity: certification of algorithm correctness and consistency; application of rules to avoid erroneous changes (in system configurations), handling both intentional and functional configuration-errors;
2) Disaster recovery and business continuity: use of fallback-routines for the handling of system outage;
3) Endpoint protection: Firewalls, Identity and access management (IAM), Intrusion prevention systems (IPS/IDS), Encryption tools, etc.

In this work we focus on addressing these aspects by reducing the expressive power, hence, reducing the number of failed states. This is motivated by the properties of mechanical systems. In mechanical systems there is a limited number of possible combinations, hence, making it necessary to support complex grammar.

C. The design of cyber-secure classification algorithms

The main challenge in the design of accurate and fast heuristics concerns the interception of tacit patterns used by human experts to deduce answers from complex data-sets. This requires a classification algorithm with the following key-features [15], [16]. Therefore, a prerequisite for safe and sound AI algorithm predictions is a strategy for capturing this

 ISBN: 978-1-61208-850-1

TABLE I
ASPECTS OF CYBER-SECURITY. THE BELOW TABLE EXEMPLIFIES HOW THE PROPOSED METHODOLOGY ADDRESSES ISSUES IN CYBER-SECURITY AND EXECUTION-TIME.

		Non-Heterogenous AI		Heterogeneous AI	
		Definition		Definition	
What	When	How	Issue	How	Benefit
Data compression	Handling of large data streams	Subsampling, data-reduction, algorithms	Trust	Partial Computing	Known variance
Limited data bandwidth	Communication between devices	Subsampling	Provenance	Partial Computing	Known Variance
Calculation of AI	Data from sensors	Assumptions of distributions	Undocumented assumptions	Semantics	Trust
Configuration of algorithms	Computers with limited resources	Simplified algorithms	Handling of Outliers	Partial Computations	Provenance, Accuracy
Use of reference data in algorithms	State changes	Guesses based on data-changes	Unawareness of changes	Updates through semantics	Pareto Boundary
Handling of Data distortions	Signalling noise in the data-cable	Generalized assumptions	Execution-time, inaccurate assumptions	Semantics, software	Provenance
Classify actors in an image	Object seen from odd angles	Training Data	Loss of inferences	Algorithmic Building Blocks	Cognitive radius
Analysis of external data-sets sampled from different data resources with unknown origin	distortion along an unexpected axis	Average normalization	Unexpected behaviour	Provenance, semantics	Flow of documentation
Analysis never completes	Data too large for microprocessors	Inaccurate algorithm	Inaccurate predictions	hpLysis software	Accurate predictions

intersection: to map cognitive (or: philosophical) perspectives of patterns to the design of fast and accurate algorithms heuristics. This implies addressing the issues of:

1) data throughput: IoT equipment are interconnected through multiple layers of networks with poorer bandwidth, as seen for Fieldbus networks [17];
2) AI-algorithms: a need for accurate and fast classification and ranking of equipment status, which may be generalized into the tasks of cluster analysis for hypothesis evaluation;
3) computing power: computers embedded on IoTs go approx 10^2x slower than the microprocessors found on sensors, and approx 10^4x slower than desktop computers [10].

The observations argue for identifying algorithms that may efficiently be applied to legacy IoTs. If successful, the approach is bound to have an impact on 200 billion+ computers [18]: the global sensor market is expected to reach $287.00 Billion by 2025 [9]. Our earlier research reveals how the cost of AI may be reduced by 100x+ while improving the trustworthiness of predictions [19]. To summarise, the task of redefining existing sensor networks into autonomous equipment requires an AI algorithm with a high degree of prediction trustworthiness and is feasible to integrate on existing Intelligent / Internet of Things (IoT) microprocessors. In the next sections, we outline the results of this strategy.

Algorithm 1 The proposed *ultraFast* algorithm.

Output: $clusters = []$

procedure TRAIN(Normalization, MergeMetrics, SimilarityMetrics, EntropyFunctions, F_e, F_sS) ▷ Task: learn how to classify data:
    Result ▷ holds the result-function
    for each $n \in Normalization$ **do**
        for each $m \in MergeMetrics$ **do**
            for each $s \in SimilarityMetrics$ **do**
                vec = []
                for each $f \in EntropyFunctions$ **do**
                    s = F_e(...) ▷ Task: reduce dimension from $data = [rows, columns]$ to $scalar$
                    vec.push(s)
            ▷ Task: identify accuracy of training-paramters:
                F_s(Result, vec, ...)
procedure CATEGORIZE(TrainedData, data) ▷ Task: Apply the $O(n)$ algorithm:
    class ▷ Holds the answer
    for each $f \in TrainedData$ **do**
        t = distance(f, data)
        if (**then**t.d < class.d)
            class = t

IV. METHOD: A NEW CLASSIFICATION $O(n)$ ALGORITHM FOR ACCURATE CLASSIFICATION

This section describes a framework for construction of a fast classification algorithm (Table I), which involves the design of

TABLE II
THE APPROXIMATE TIME COMPLEXITY OF CLUSTER ALGORITHMS (SUBSET). IN THIS TABLE n DENOTES THE NUMBER OF FEATURE ROWS, f IS THE NUMBER OF FEATURES, c IS THE NUMBER OF CATEGORIES AND I DENOTES THE MAXIMUM ITERATIONS.

	Time Complexity:	Relative Time [x] for n=1000	Relative Time [x] for n=1000,000
Proposed: classification (Section IV):	O(n*f)	$1x$	$1x$
KD-TREE [20]:	O(f* n log(n))	$10x$	10^4x
DBSCAN and HP-CLUSTER [21], [22], [23]:	$O(n^2 * f)$	10^4x	10^7x
Hierarchical Cluster Algorithms:	$O(n^2 * f)$	10^4x	10^7x
Kruskals MST [24]:	$O(n^2 * f)$	10^4x	10^7x
K-means [25]:	$O(n^2 * f + I * c * n * f)$	10^4x	10^7x
SOM [26]:	$O(n^2 * f + I * c^2 * n)$	10^4x	10^7x
Neural Networks []:	$O(f * n^5)$	$1000^4x = 10^7x$	$10^{10}x$

an algorithm where:

1) Approximate Computing: subsection IV-A identifies a strategy to transform existing algorithms (which makes use of multiple centroids, or: neurons) into a problem requiring a single neuron;
2) execution-time: subsection IV-B describes an algorithm which turns the observation from subsection IV-A into an $O(n)$ algorithm;
3) accuracy and provenance: subsection IV-C exemplifies how the $O(n)$ algorithm (subsection IV-A) applied to image classification.

A. Problem transformation: application of Least Parsimony to Neural Networks

To reduce the execution time of algorithms, it is of importance to minimize the number of dimensions to evaluate. Hence, to transform the evaluation problem through the principle of Least Parsimony [15]. The idea is to compute entropy by taking the distance from each midpoint to each color. SOM organises the points based on similarities in RGB. The work of [27] applies SOM to construct a two dimensional scheme of entropy computations (Eq. 1):

$$signature = \sum_{x \in C} min(C_k, d(x, C_k))\, x \in C_k \qquad (1)$$

where C_k denotes data-rows in cluster k, $d(x, C_k)$ is the feature similarity between the cluster versus the data-row x, and where C holds the clusters, while $|C|$ holds the set of all data-rows (*e.g.*, in the input image). From Eq. 1 we observe how prediction inaccuracies arise when the *within-distance* is not significantly greater than the *between-distance* (Eq. 2):

$$\sum_{x \in C} min_{k \in C}(min(d(x, C_k))) < \textit{Eq. 1} \qquad (2)$$

For cases where the *between distance* is smaller than the *within distance* the splitting of points between clusters becomes pointless (2), *i.e.*, as the prediction specificity is not improved. Hence, when SOM is applied for data outside the algorithms Pareto Boundary then using multiple centroids (or: neurons) increases the prediction error rate. This exemplifies how costly algorithms may be redesigned into the use of a single reference point, where the latter becomes equivalent to the direct use of entropy metrics.

B. An $O(n)$ algorithm for classification

The motivation is to design an effective algorithm for classification. This algorithmic learning-phase can be generalized into:

1) ensemble data: a list of ranked (*i.e.*, ordered) data; used to determine in the *selection phase* to determine the best-performing *algorithm permutation*;
2) algorithm permutation: uses a selection of building blocks to construct a pipeline of algorithm-training; the iterative sequence (of this feature-scaling) ensures that the identified algorithm has a time complexity of $O(n)$ for n data-points;
3) selection phase: each *algorithmic perturbation* produce a scalar number (*e.g.*, number=2.0001); the number is inserted into a vector; when all the data-sets (in a data-ensemble) are calculated, the vector is compared to the expected order (of data, as defined in the *enamdale data* phase);

The above steps are formalised into an algorithm for value selection (Alg. 1). The following subsection IV-C exemplifies how Alg. 1 can be trained for image-classification, a task of higher complexity than classification of sensor-data.

C. Automated Algorithm Configuration: training and evaluation

To train the algorithm, we provide a tool-suite for the exploration of algorithm combinations, and templates for mapping the properties into implementation with low execution-time and small assembly instruction size. Therefore, the approach may be used for the training of algorithms applicable to legacy IoT microprocessors. An example is to apply an automated evaluation strategy considering the building blocks of:

1) entropy metric: explore 20+ metrics for capturing the variance in a distribution of numbers;
2) down-sampling: condense numbers through compression, for which blocks of adjacent numbers are constructed;
3) blurring: include perspective provided by each number through brushing, for which we explore the combinations of *unchanged*, use a linear attenuation threshold, etc.;
4) strategy for converting input image to histogram: none, bins=[10, 100, 1000] x [raw, average, sum];

 ISBN: 978-1-61208-850-1

5) RGB to scalar conversion: translate the "Red, Green, Blue" scores in images into a singular channel (*e.g.*, Hue);
6) normalization: explore the effect of normalization values through different combinations of the midpoint (*e.g.*, the value of averaged score), signed, etc.;
7) combine data: determine how gold hypothesis is to be used, *e.g.*, to merge features based on relationships such as: multiply, (maximum/minimum), etc.;
8) pairwise similarity metric: apply the 320+ metrics [28].

The results provide a proof of concept for the assertion that entropy-algorithm supports *re-invention*, *e.g.*, that it manages to get results at-least-as-good as the SOM-method. The comparison of data with a known topology avoids the need for complex iteration steps (Figure 1), which is in contrast to other algorithms(eg as "SLINK" [29], "k-means" [25], etc.). Hence, explaining why the proposed framework enables a reduction in execution time by 10^4+ for data-set with 1000 points (Table II).

V. Results

The findings provide insight into the feasibility of transforming Neural Networks into the Ultra-fast $O(n)$ algorithms (Alg. 1). The idea (which is explored) is to use a singular centroid to capture complexities (subsection IV-A), for which a problem is rewritten through use of algorithmic building blocks. The proposed algorithm enables fast and secure communication over insecure networks: by reducing the processing-time, and amount of data to transfer, users are able to apply cryptography strategies (an overhead would otherwise be unbearable). To validate the feasibility of the proposed guidelines we explore:

1) accuracy and provenance: to measure the algorithm's feasibility, we investigate the algorithm through generic, and specific, use-cases, *e.g.*, the accuracy of image-classification;
2) execution-time: to identify any overhead in execution-time, the hpLysis software [30] is updated with Alg. 1, where results are summarized in Table II;
3) approximate computing: to explore the effects, this paper transform a set of complex classifications into a simple comparison (Figure 1) through the use of Alg. 1 (Eq. 1), and then explores the difference in performance.

The results are summarized in Table II, which identifies the relative execution-time for different algorithms. Winns (a producer of heat-exchange pumps [1]) reports that sensor-predictions need to be returned in less than 10 seconds, which only Alg. 1 manages (Table II). When Alg. 1 is evaluated through the above perspectives we observe how Alg. 1 outperforms the base-line algorithms in use:

1) Specific application: classification of image-data, here exemplified through the Las Vegas data-set and the Lake Mead dataset found in [31];
2) Signal classification: classify shapes with different growth-ratios and ranomdation, *i.e.*, $y(r, a, x) = r_1 a_1 x^0 + r_2 a_2 x^2 + ... r_n a_n x^n$, where $y(..)$ is the feature-vector to evaluate, n is the number of combinations (to construct a signal from), r is a constant randomization-factor, a is a constant attenuation-factor (*e.g.*, $a = 1.5$), and x represents the polynomial variabel-part;
3) Generalized applicability: the hpLysis is updated with generalized tests, each investigating the effects of Approximate Computing on Neural Network.

To exemplify, a comparison between SOM-strategy for low-latency classification (undertaken by [27], [31]) versus Alg. 1 (as proposed in Section IV) indicates the transformation of algorithms into using a singular centroid (Eq. 1) can substantially boost the performance of analytical approaches. Discussions with the authors of [31] reveal how data-specific configurations of the SOM are required to get the algorithm to produce correct results. The results reveal how the use of a singular centroid (in data-classification) provides a simple, yet effective, strategy for trustworthy control of classification-tasks. Therefore, the algorithm may readily be used on existing sensor networks, *e.g.*, to control equipment for heat-exchange in buildings.

VI. Discussion

This paper has identified a low-cost method to classify data with well-defined characteristics, which is the case for sensors that monitor physical equipment (Table I). To reduce the scope, the paper focuses on industrial control-systems which a) are sensitive to delays in configurations, and b) where certification of behavior represents a crux. The paper argues that a holistic perspective of classification-algorithms results in a cost-effective strategy to address issues in data-throughput. The proposed methodology, and algorithm, differ from the established strategy. To exemplify:

1) Approximate Computing: this work explores the benefit of closely gluing compiler-optimization with the accuracy of algorithms, *e.g.*, in contrast to "scikit learn" [32];
2) execution-time: we transform complex algorithms (into their simplified counterparts) by merging the cluster-centroids, *e.g.*, in contrast to [31];
3) accuracy and provenance: the use of metric-training (Alg. 1) a) relates to a system's physical properties, and b) captures the algorithmic behavior, *e.g.*, in contrast to [33].

This work exemplifies a methodology that is generalizable for a wider audience; through an intersection between established algorithms, use-cases, and configurations, the paper reveals a strategy reducing the execution-time by more than 10,000x. The paper argues that the approach can be applied to arbitrary cases of classification, such as the classification of sensor data from IoT, a ranking of satellite images, etc. A concrete example concerns the effects of the accurate choice of pairwise similarity metrics in the clustering algorithm.

VII. Conclusion

The paper proposes a parametric strategy to increase the applicability of classification algorithms: observations relating

 ISBN: 978-1-61208-850-1

to the approximate nature (of classification algorithms) are used to derive a new $O(n)$ algorithm. The algorithm is both evaluated through generalized, and highly specific, datasets, hence ensuring its broad applicability. The use of well-defined metrics, reflecting the physical properties of sensor-systems, makes the algorithm easy to certify: the seamless use of off-the-shelf building blocks address issues in data-throughput, the trustworthiness of predictions, and the speed of microprocessors, *i.e.*, without resulting in increased component costs. Thereby, the paper provides a template addressing the daunting challenges facing researchers, managers, and owners of industrial systems. Hence, the proposed algorithm addresses the conceptual challenges which currently hampers the development of trustworthy applications of AI to the autonomous control of industrial systems.

The findings presented in this paper indicate the need for updating the requirements for the certification of sensors and equipment. Hence, there is a need for a concerted effort in the industry, *i.e.*, to devise a formal protocol that ensures flexible and safe AI for industrial sensor networks.

REFERENCES

[1] E. Morset, "Email conversations with the cto of winns reveals how accurete regulations of heat-pumps maps to their energy consumption." Winns, 2021, accessed: January 2021.

[2] S. Liu, B. Xing, B. Li, and M. Gu, "Ship information system: overview and research trends," International Journal of Naval Architecture and Ocean Engineering, vol. 6, no. 3, 2014, pp. 670–684.

[3] E. McMahon, M. Patton, S. Samtani, and H. Chen, "Benchmarking vulnerability assessment tools for enhanced cyber-physical system (cps) resiliency," in 2018 IEEE International Conference on Intelligence and Security Informatics (ISI). IEEE, 2018, pp. 100–105.

[4] US-CERT, "Alert (ta16-288a) heightened ddos threat posed by mirai and other botnets," 2016, accessed: September 2019. [Online]. Available: https://www.us-cert.gov/ncas/alerts/TA16-288A

[5] K. Q. Ye, M. Green, N. Sanguansin, L. Beringer, A. Petcher, and A. W. Appel, "Verified correctness and security of mbedtls hmac-drbg," in Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. ACM, 2017, pp. 2007–2020.

[6] J. H. Ziegeldorf, O. G. Morchon, and K. Wehrle, "Privacy in the internet of things: threats and challenges," Security and Communication Networks, vol. 7, no. 12, 2014, pp. 2728–2742.

[7] B. Xing, J. Dai, and S. Liu, "Enforcement of opacity security properties for ship information system," International Journal of Naval Architecture and Ocean Engineering, vol. 8, no. 5, 2016, pp. 423–433.

[8] D. Trendafilov, K. Zia, A. Ferscha, A. Abbas, B. Azadi, J. Selymes, and M. Haslgrübler, "Cognitive products: System architecture and operational principles," 2019.

[9] Bloomberg, "Sensor market estimated to reach 287 billion globally by 2025," 2019, accessed: September 2019.

[10] T. Adegbija, A. Rogacs, C. Patel, and A. Gordon-Ross, "Microprocessor optimizations for the internet of things: A survey," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 37, no. 1, 2017, pp. 7–20.

[11] O. K. Ekseth, M. Gribbestad, and S.-O. Hvasshovd, "Inventing wheels: why improvements to established cluster algorithms fails to catch the wheel," in The International Conference on Digital Image and Signal Processing (DISP19), Springer, 2019.

[12] O. K. Ekseth, J. C. Meyer, and S. O. Hvasshovd, "hplysis database-engine: A new data-scheme for fast semantic queries in biomedical databases," in Semantic Computing (ICSC), 2018 IEEE 12th International Conference on. IEEE, 2018, pp. 383–390.

[13] L. Ana and A. K. Jain, "Robust data clustering," in Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on, vol. 2. IEEE, 2003, pp. II–128.

[14] T. Economist, "War in the fifth domain: Are the mouse and keyboard the new weapons of conflict?" 2010, accessed: June 2020. [Online]. Available: https://www.economist.com/briefing/2010/07/01/war-in-the-fifth-domain

[15] B. Dresp-Langley, O. K. Ekseth, J. Fesl, S. Gohshi, M. Kurz, and H.-W. Sehring, "Occams razor for big data? on detecting quality in large unstructured datasets," Applied Sciences, vol. 9, no. 15, 2019, p. 3065.

[16] O. K. Ekseth, P.-J. Furnes, and S.-O. Hvasshovd, "Pattern matching in the era of big data: A benchmark of cluster quality metrics." International Journal On Advances in Software, 2019.

[17] A. Pietak and M. Mikulski, "On the adaptation of can bus network for use in the ship electronic systems," Polish Maritime Research, vol. 16, no. 4, 2009, pp. 62–69.

[18] Intel, "200 billion iot devices in 2020, a market values to $6.2 trillion in 2025," 2019, accessed: December 2019. [Online]. Available: https://www.intel.com/content/dam/www/public/us/en/images/iot/guide-to-iot-infographic.png

[19] O. K. Ekseth and S.-O. Hvasshovd, "An empirical study of strategies boosts performance of mutual information similarity," in International Conference on Artificial Intelligence and Soft Computing. Springer, 2018, pp. 321–332.

[20] J. H. Friedman, J. L. Bentley, and R. A. Finkel, "An algorithm for finding best matches in logarithmic expected time," ACM Transactions on Mathematical Software (TOMS), vol. 3, no. 3, 1977, pp. 209–226.

[21] M. Ester, H.-P. Kriegel, J. Sander, X. Xu et al., "A density-based algorithm for discovering clusters in large spatial databases with noise." in Kdd, vol. 96, no. 34, 1996, pp. 226–231.

[22] O. K. Ekseth and S. Hvasshovd, "hplysis dbscan: How a memory-aware db-scan implementation out-perform simplified/heuristic db-scan approaches by 1,000,000x+," 2017, pp. 1–6.

[23] O. K. Ekseth and S.-O. Hvasshovd, "hp-cluster: A new algorithm enables increased performance for clustering of big, and complex, data," 2020, manuscript ready for submission.

[24] J. B. Kruskal, "On the shortest spanning subtree of a graph and the traveling salesman problem," Proceedings of the American Mathematical society, vol. 7, no. 1, 1956, pp. 48–50.

[25] S. Lloyd, "Least squares quantization in pcm," IEEE transactions on information theory, vol. 28, no. 2, 1982, pp. 129–137.

[26] T. Kohonen and P. Somervuo, "Self-organizing maps of symbol strings," Neurocomputing, vol. 21, no. 1, 1998, pp. 19–30.

[27] J. M. Wandeto and B. Dresp, "Ultrafast automatic classification of sem image sets showing cd4 + cells with varying extent of hiv virion infection." International Journal On Advances in Software, 2019.

[28] O. K. Ekseth and S.-O. Hvasshovd, "How an optimized DBSCAN implementation reduce execution-time and memory-requirements for large data-sets." International Journal On Advances in Software, 2017, pp. 321–332.

[29] R. Sibson, "Slink: an optimally efficient algorithm for the single-link cluster method," The computer journal, vol. 16, no. 1, 1973, pp. 30–34.

[30] Ekseth, Ole Kristian, "hpLysis: a high-performance software-library for big-data machine-learning," https://bitbucket.org/oekseth/hplysis-cluster-analysis-software/, online; accessed 06. Jan. 2021.

[31] J. M. Wandeto and B. Dresp-Langley, "The quantization error in a self-organizing map as a contrast and colour specific indicator of single-pixel change in large random patterns," Neural Networks, vol. 119, 2019, pp. 273–285.

[32] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg et al., "Scikit-learn: Machine learning in python," Journal of Machine Learning Research, vol. 12, no. Oct, 2011, pp. 2825–2830.

[33] D. Moulavi, P. A. Jaskowiak, R. J. Campello, A. Zimek, and J. Sander, "Density-based clustering validation," in Proceedings of the 2014 SIAM International Conference on Data Mining. SIAM, 2014, pp. 839–847.

 ISBN: 978-1-61208-850-1

Extraction of News Articles Related to Stock Price Fluctuation Using Sentiment Expression

Kazuto Tanaka
Major in Computer and Information Sciences
Graduate School of Science and Engineering,
Ibaraki University
email: 20nm715g@vc.ibaraki.ac.jp
4-12-1, Nakanarusawa, Hitachi, Ibaraki, Japan

Minoru Sasaki
Dept. of Computer and Information Sciences
Faculty of Engineering, Ibaraki University
email: minoru.sasaki.01@vc.ibaraki.ac.jp
4-12-1, Nakanarusawa, Hitachi, Ibaraki, Japan

***Abstract*—In recent years, various economic reports are published and they are important tools in all markets. However, it is necessary to consult a lot of news articles in order to write an economic report. In this paper, we propose a method for effectively extracting news articles including important events related to the fluctuation of the Nikkei Stock Average by using Japanese sentiment lexicons. The results of the experiments show that the proposed method reduces the number of articles by about half and retrieves relevant documents better than the method using only stock price fluctuations. Therefore, the Japanese sentiment lexicons is effective for extracting news articles including important events.**

Keywords- economic articles extraction; sentiment analysis; Nikkei Stock Average

I. INTRODUCTION

In today's information society, many individual investors try to stay up to date with the latest situations by reading news articles on the web. The news articles describe happenings that are currently occurring in the world and contain information valuable for various purposes. Thus, it is important for investors and traders to be aware of current market situations. However, with the exponentially increasing amount of available news articles, it has become a critical matter to utilize the information from these data in decision making process.

There are various approaches to perform event detection in news articles to help analysts to analyse large amounts of financial news articles. For example, Nakayama et al. propose a method that adds the words appearing in the article and the words appearing in the web page of the company, and uses these words as input for Support Vector Machine (SVM) [1]. In addition, Sakai et al. proposed a method to extract causal information as a causal expression using statistical information and initial cue expressions [2] , and Milea et al. proposed a method using fuzzy grammars [3]. In this existing approaches, we focus on research that analyses the relationship between stock price fluctuations and news articles using text mining techniques. Valuable and significant information related to the financial markets is now available easily on the Internet. Hence, it is neither insignificant nor simple to obtain valuable information and analyse the relationship between the information obtained and the financial markets. However, some existing studies have automatically generated reports by extracting important events and performance factors from previously extracted important news articles. Therefore, there were few studies that automatically extracted important news articles from a large number of news articles.

In this paper, we propose a method to extract important articles related to the fluctuation of the Nikkei Stock Average by using Japanese sentiment lexicons from financial news articles. In the proposed method, we show that the use of the Japanese sentiment lexicons is effective for extracting useful news articles. By using the proposed method, it is possible to automatically extract useful articles for making reports from a large amount of news article data.

The rest of this paper is organized as follows. Section 2 is devoted to the related works in the literature. Section 3 describes the proposed important news extraction method. We describe an outline of experiments in Section 4 and experimental results in Section 5. Finally, we discuss the results in Section 6 and concludes the paper in Section 7.

II. RELATED WORKS AND METHODS

This section presents the related works and methods related to our research.

A. Related Works

In recent years, there has been increasing interest in applying neural networks and machine learning techniques to solve this task. First, we introduce some researches related to the analysis of fluctuation of the Nikkei Stock Average.

Nakayama et al. used the Nihon Keizai Shimbun to extract articles that describe events that affect the stock prices of some selected companies [1]. In this paper, they propose a method that adds the words appearing in the article and the words appearing in the web page of the company, and uses these words as input for SVM. This method improved the accuracy by 27.2% compared to the method before adding the words in the web page. However, this experiment does not take into account the combination with the Nikkei Stock Average.

Sakai et al proposed to use Japanese financial news to extract information on the causes of fluctuations in corporate performance [2]. In this paper, causal information is extracted as causal expressions using statistical information and initial cue expressions. As a result, the accuracy of the method was 79.2%, which is better than the result of the conventional method.

Milea et al, proposed to predict the movement of the MSCI Euro Index from the report prepared by the European Central Bank (ECB) [3]. In this paper, they used a fuzzy grammar to

 ISBN: 978-1-61208-850-1

create their model. As a result, the results are the same as in previous studies, but the model is simplified.

As mentioned above, there have been many studies on predicting and extracting the causes of fluctuations from various texts, but few studies have focused on extraction of important articles related to the stock price fluctuation to generate investor reports. In this study, we focused on the extraction of important articles related to the Nikkei Stock Average.

Next, we introduce some researches related to the financial text analysis based on sentiment dictionary.

Sato et al. investigated the correlation with stock price fluctuations using a sentiment analysis for news articles in Japan [4]. The results showed that all correlations were low and no correlation could be found between them.

Yazdani et al. proposed a classification method to classify financial news articles into positive or negative class [5]. In their experiments, they used the N-gram models unigram, bigram, and a combination of unigram and bigram as feature extraction, and used Document Frequency (DF) to evaluate the N-gram models and traditional feature weighting methods. The results show that the feature selection and feature weighting methods play a significant role in the negative-positive classification of articles.

Yadav et al. proposed an unsupervised learning model for sentiment classification of financial news articles [6]. The experiment was conducted by using POS based feature extraction with seed set and noun-verb combination in the traditional method. The results showed that using the noun-verb combination was better than the traditional method.

With reference to the above studies, we decided to focus on sentiment expressions.

B. Sentiment Expression

There are two types of words: those that give a good impression (positive) and those that give a bad impression (negative). These words are expressed by converting them into numerical values, which is called sentiment information. One of the ways to convert sentiment information is to use a sentiment dictionary. A sentiment dictionary is a dictionary of values determined for each word. The sentiment dictionary we used this time was provided by the Okumura-Takamura Laboratory at Tokyo Institute of Technology [7]. The sentiment dictionary is expressed using numbers between 1 and -1, where the closer the number is to 1, the more positive the word is, and the closer the number is to -1, the more negative the word is. In this study, we use this method to convert news articles into numerical values.

C. Morphological Analysis

Unlike other languages, Japanese is characterized by the fact that words are not separated, but are written consecutively. Therefore, it is necessary to divide a sentence into separate words. Morphological analysis is a technology that allows a machine to do this automatically.

In this study, we used a morphological analysis package called janome. This janome uses dictionaries used by a morphological analysis engine called MeCab, which can perform morphological analysis with high accuracy, so it can perform morphological analysis with the same high accuracy as MeCab.

In addition, these morphological analysis engines can not only segment each word, but also analyse the parts of speech and conjugations of the segmented words.

Figure 1. Examples of Morphological Analysis

As an example, the morphological analysis of "私は明日東京へ出発します"(I am leaving for Tokyo tomorrow) is shown in Figure 1.

III. Extraction Method Using Sentiment Expression

In this section, we present a method to extract news articles that have information related to the fluctuation of the Nikkei Stock Average.

A. Overview of the Proposed Method

Figure 2 shows a rough sequence of the proposed method.

"Headline" in the figure is the title of the news article. "Date" is the date when the news article was published. "Open" is the opening price of the Nikkei Stock Average. "Close" is the closing price of the Nikkei Stock Average. "Positive", "neutral", and "negative" are numerical values for each word in the headline. "Positive score", "neutral score", and "negative score" are each summarized for each day.

First, we select data from article data and stock price data. Since the article data contains some articles that may be noisy, we select articles under certain conditions. The details are introduced in Section B. We considered the correct article to be an article about a major event related to the fluctuation of the Nikkei Stock Average.

Next, the selected articles are polarity transformed. Before polarity transformation, it is necessary to divide the keywords of the articles into the smallest units of words. Therefore, the keywords are first analysed by morphological analysis, and then converted into numerical values by the polarity dictionary. The detailed conversion method will be explained in Section C.

Then, we use the obtained polarity to calculate whether the article is a positive or negative article on stock prices. The method is to calculate the three percentages of whether the

 ISBN: 978-1-61208-850-1

article is positive, negative, or irrelevant (neutral). The detailed calculation method will be explained in Section D.

Finally, we extract articles from the results of polarity representation. Articles with positive values in the top 75% and negative values in the bottom 25% are extracted. At this time, if duplicates often occur in both values, the duplicates are removed.

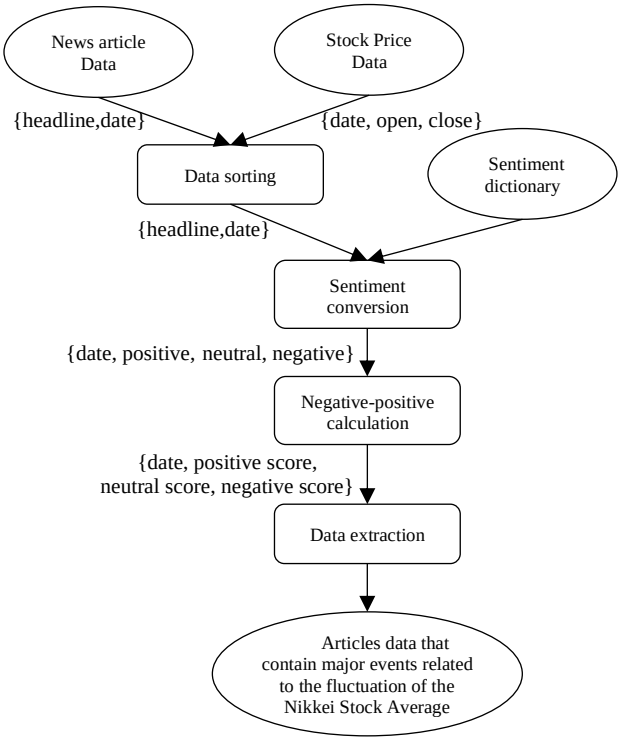


Figure 2. Flow of the Proposed Method

We compare the percentage of articles that contain major events related to the fluctuation of the Nikkei Stock Average between the extracted set of articles and the unextracted set of articles to verify the effectiveness of sentiment expression.

B. Preprocessing

First of all, we need to sort out the data from Nikkei QUICK News, which we use, because there are some news articles that are not related to the Nikkei Stock Average, such as sports articles. The first step is to limit the news articles to two sources, "QUICK" and "NQN". Since these two sources only provide news about stocks, we limited ourselves to these two sources.

Next, news articles distributed by "QUICK" and "NQN" include a mixture of news articles summarizing the past and news articles about a company's sales. These articles can have a negative impact on the use of sentiment expressions. Therefore, we removed such articles.

In addition, we considered that news articles related to the fluctuation of the Nikkei Stock Average were likely to be transmitted on the days when major fluctuations occurred. Therefore, we limited ourselves to the news articles that were published on days when the Nikkei Stock Average was more than 1% or less than -1% compared to the previous day. In this paper, such days are referred to as Fluctuation Day.

Finally, in some of the data used in this study, the keywords of the news articles are empty. In this paper, the articles with empty keywords are also removed in order to use these keywords for polarity conversion.

C. Sentiment Conversion

Sentiment conversion of articles limited by preprocessing into three categories: positive, negative, and neutral. Neutral means a word that is neither positive nor negative. For this conversion, we morphologically analyse the keywords in the news article. This is because the words present in the sentiment dictionary are the smallest words registered in natural language. The words obtained from this morphological analysis are compared with the sentiment dictionary and converted into numerical values. Positive values are considered positive, and negative values are considered negative. If the word does not exist in the sentiment dictionary, it is considered a neutral word and converted to 0. If the word is not in the sentiment dictionary, it is considered a neutral word and converted to 0. Furthermore, words that are replaced with 0 when converted to numerical values are treated as neutral words. This is done for all articles.

D. Negative-Positive Calculation

Each of the positive, negative, and neutral values converted in the previous section are combined into one. In this experiment, we averaged the positive, negative, and neutral values for each article and used the individual values to find the percentage of each. The reason for using percentages is that the number of keywords in each article is different, and it is easy to extract articles with many keywords using only the average. Therefore, this time we will use the average value to obtain the percentage. Specifically, we used the formula below to calculate the percentage. Since neutral words are less important than positive and negative words, the average of all neutral words is set to 1 if neutral words exist, and 0 otherwise.

- Positivity formula

$$positive\_score = \frac{\overline{positive}}{\overline{positive} + \overline{negative} + \overline{neutral}}$$

- Negatives formula

$$negative\_score = \frac{\overline{negative}}{\overline{positive} + \overline{negative} + \overline{neutral}}$$

 ISBN: 978-1-61208-850-1

- Neutral formula

$$neutral\_score = \frac{\overline{neutral}}{\overline{positive} + \overline{negative} + \overline{neutral}}$$

IV. EXPERIMENTS

The experimental procedure based on the proposed method is shown.

A. Data Set

In this study, we used data compiled from Nikkei QUICK News for 2016 and 2017 and the prices of the Nikkei Stock Average for 1747 days from 2011 to 2017, as well as a polar dictionary provided by Okumura and Takamura laboratories at Tokyo Institute of Technology [7].

B. Settings

In this experiment, we decided to focus on three important events: the Ise-Shima Summit, the U.S. presidential election, and the North Korean missile launch. Therefore, the experiment was conducted every month in the month in which these events occurred. In doing so, articles that included "サミット"(summit), "選挙"(election), and "地政学リスク"(geopolitical risk) as keywords in their articles were considered as correct articles.

- May 2016: Ise-Shima Summit
- November 2016: U.S. presidential election
- August 2017: geopolitical risk

In addition, the U.S. is often involved in the fluctuations of the Nikkei Stock Average. Therefore, we conducted an experiment with an article set that was limited to articles that contained "米国"(U.S.) or "アメリカ"(America) in the keywords of the articles. In this study, we refer to this set of articles as the set of articles containing "U.S.".

Thus, this study was conducted using two patterns of data for each of the three events.

V. RESULTS

The results of an experiment conducted based on the experimental method are shown below.

A. Results of the May 2016 Article Set

The results done in May 2016 are shown in Table 1 below. Also, there are six Fluctuation Days: 2, 10, 13, 17, 25, and 30.

TABLE I. RESULTS OF THE MAY 2016 ARTICLE SET

Article set conditions	Total correct articles	Total articles	percentage(%)	rise in value
F.D	55	2274	2.419	
F.D +senti	26	1045	2.488	0.069
F.D +U.S.	17	310	5.484	
F.D +U.S. +senti	10	156	6.410	0.926

In the table, "F.D" refers to Fuctuation Day. Also, "F.D + U.S." refers to the set of articles that include "U.S.". In addition, "F.D +senti" and "F.D +U.S. +senti" are the results of extraction using sentiment expressions. "Total correct articles" is the number of correct articles for each condition. "Total articles" is the number of articles when the data is sorted under each condition. "Percentage" is the ratio of correct articles to the total number of articles. "Rise in value" is the increase in percentage after the experiment.

B. Results of the November 2016 Article Set

The results done in November 2016 are shown in Table 2 below.

TABLE II. RESULTS OF THE NOVEMBER 2016 ARTICLE SET

Article set conditions	Total correct articles	Total articles	percentage(%)	rise in value
F.D	112	2890	3.875	
F.D +senti	51	1337	3.815	-0.060
F.D +U.S.	49	463	10.583	
F.D +U.S. +senti	25	231	10.823	0.240

It is the same as Table 1 with respect to the making of the table. In addition, there are seven Fluctuation Days: 2, 4, 7, 9, 10, 14, and 16.

C. Results of the August 2017 Article Set

The results done in August 2017 are shown in Table 3 below.

TABLE III. RESULTS OF THE AUGUST 2017 ARTICLE SET

Article set conditions	Total correct articles	Total articles	percentage(%)	rise in value
F.D	50	941	5.313	
F.D +senti	29	432	6.713	1.400
F.D +U.S.	17	155	10.968	
F.D +U.S. +senti	11	78	14.103	3.135

It is the same as Table 1 with respect to the making of the table. In addition, there are six Fluctuation Days: 9, 15, and 18.

VI. DISCUSSIONS

In this section, we discuss the experimental results and show the effectiveness and problems of polarity representation.

A. Validity of Sentiment Expression

The results of Section 5 show that all article sets have a better percentage of articles containing information related to the fluctuation of the Nikkei Stock Average, except for the extraction performed on the article set for the Fluctuation Day of November 2016. In addition, this extraction method

 ISBN: 978-1-61208-850-1

reduced the original article set by almost half, which means that we were able to reduce the number of articles that need to be manually examined.

The reason for the worse results in November 2016 was that there were articles on Clinton's email problem and articles on the current status of the presidential election that did not include "選挙"(election) as a keyword. Therefore, it is thought that such articles were extracted rather than articles containing "選挙"(election) and thus the good results were not obtained.

Considering the above, we believe that the use of sentiment expressions is effective in extracting articles that have information related to the fluctuation of the Nikkei Stock Average.

B. Inadequate Preprocessing

From the results of Section 5, we can see that both the percentage and the upward value of the set of articles including "U.S." are better than the set of Fluctuation Day. This was probably due to the fact that we were able to remove articles that were not so relevant to the fluctuation of the Nikkei Stock Average, such as news about companies, since the percentage was better just by limiting the articles to those that included "U.S.". Therefore, the pre-processing of this experiment alone is not enough to sort out the data.

In the future, it is necessary to review the data selection process and to improve the weighting of each article.

C. Optimal Threshold for Front-to-back Comparison

This study was limited to the days when the Nikkei Stock Average was more than 1% or less than -1% compared to the previous day. Therefore, differences occur in the number of days examined in different months. In this study, there were six days in May 2016 and seven days in November 2016, but in August 2017, there were only three days, less than half. Furthermore, in August 2017, only the 9th, 15th, and 18th were Fluctuation Day, so it was not possible to take articles from the beginning and end of the month. Therefore, it is necessary to adjust this threshold of the Nikkei Stock Average compared to the previous day from month to month.

VII. CONCLUSION

In this study, news articles related to the fluctuation of the Nikkei Stock Average were extracted from the price information of the Nikkei Stock Average and Nikkei QUICK News by using sentiment expressions. The news articles were converted into numerical values from the sentiment dictionary, and the articles were extracted from these values. To verify the effectiveness of the extraction method using sentiment expressions, we compared the percentage of news articles related to the fluctuation with the original set of articles. As a result, the number of articles in the article set was reduced by almost half when the extraction method using sentiment expressions was used, and the percentage of articles included in the articles related to fluctuations also increased. Thus, we were able to confirm the effectiveness of polar expressions.

Future tasks include improving the pre-processing data selection method, weighting of each article, and adjustment of the fluctuation date threshold for each month.

REFERENCES

[1] M. Nakayama, H. Sakaji, and K. Katsuta, "Hiroyuki Sakai: Extraction of Important Articles that Influence the Stock Price of Companies from Financial Articles," The Journal of the Faculty of Science and Technology, Seikei University, Vol.51, No.2, pp.53-60, 2014.

[2] H. Sakai, and S. Masuyama, "Cause Information Extraction from Financial Articles Concerning Business Performance," IEICE Transactions on Information and Systems, Vol.E91-D, No. 4, pp. 959–968, 2008

[3] V. Milea, N. M. Sharef, R. J. Almeida, U. Kaymak, and F. Frasincar, "Prediction of the MSCI EURO index based on fuzzy gram- mar fragments extracted from European Central Bank statements," International Conference of Soft Computing and Pattern Recognition, pp.231–236, 2010.

[4] K. Sato, T. Odaka, J. Kuroiwa, and H. Shirai, "Study on the Correlation between Stock Price and Web Data Using Negative-Positive Analysis," Memoirs of the Graduate School of Engineering, University of Fukui, Vol. 63, pp.75-86, 2015.

[5] S. F. Yazdani, M. A. A. Murad, and N. M. Sharef, "Sentiment Classification of Financial News Using Statistical Features," International Journal of Pattern Recognition and Arti¯cial Intelligence Vol. 31, No. 3, 2017.

[6] A. Yadav, C. K. Jha, A. Sharan, and V. Vaish, "Sentiment analysis of financial news using unsupervised approach," Procedia Computer Science Volume 167, pp.589-598, 2020.

[7] H. Takamura, T. Inui, and M. Okumura, "Extracting Semantic Orientations Using Spin Model," Journal of Information Processing Society of Japan, Vol.47 No.02, pp. 627-637, 2006.

 ISBN: 978-1-61208-850-1

Visual Social Signals for Shoplifting Prediction

Shane Reid
School of Computing, Engineering, and intelligent systems
Ulster University
Derry/Londonderry, UK
Reid-S22@ulster.ac.uk

Sonya Coleman
School of Computing, Engineering, and intelligent systems
Ulster University
Derry/Londonderry, UK
sa.coleman@ulster.ac.uk

Philip Vance
School of Computing, Engineering, and intelligent systems
Ulster University
Derry/Londonderry, UK
p.vance@ulster.ac.uk

Dermot Kerr
School of Computing Engineering and intelligent systems
Ulster University
Derry/Londonderry, UK
d.kerr@ulster.ac.uk

Siobhan O'Neill
School of Computing, Engineering and intelligent systems
Ulster University
Derry/Londonderry, UK
sm.oneill@ulster.ac.uk

***Abstract*—Retail shoplifting is one of the most prevalent forms of theft, estimated to cost UK retailers over £1 billion in 2018. One security measure used to discourage shoplifting is surveillance cameras. However, evidence shows that unless these cameras are constantly monitored, they are ineffective. Automated approaches for detecting suspicious behaviour have proven effective but lack the transparency to enable them to be used ethically in real life scenarios. One way to overcome these problems is through the use of social signals. These are observable behaviours which can be used to predict an individual's future behaviour. To this end we have developed a set of 15 visual attributes which can be used for shoplifting prediction. We then demonstrate the effectiveness of these attributes by deriving a new dataset of visual social signals attributes by manually annotating videos from the University of central Florida Crimes dataset.**

Keywords—Social signal processing; Ethical AI; Activity recognition; Behaviour recognition; Data analytics.

I. INTRODUCTION

In 2018, retail shoplifting accounted for over £1 billion in losses for retailers in the United Kingdom [1]. In order to reduce these losses, many retailers are applying increased security measures, such as hiring security staff and using security tags on their more expensive items. The use of surveillance cameras is one method which has proven effective at deterring potential thieves. However, in order to be fully effective, these cameras need to be carefully monitored at all times [2]. Furthermore, evidence has shown that serial shoplifters have developed methods to evade security cameras, such as concealing the goods while in surveillance blind spots. This can make it very difficult for a human observer to catch all incidences of shoplifting [3].

Recently there have been attempts to automate the detection of individuals who are likely to shoplift through the use of computer vision techniques [4]. While these approaches have shown great accuracy, they are often based on black box learning techniques. This makes it impossible to justify why an individual has been classified as a potential shoplifter and raises ethical questions about how these methods come to their decision [5]. The Committee of Experts on Internet Intermediaries (MSI-NET) at the council of Europe has already outlined concerns around the admissibility of black box algorithms in criminal justice, and there are ongoing questions about the potential human rights violations of using evidence from these systems in a court of law [6].

Psychological and criminology literature has shown that individuals who are likely to shoplift exhibit a number of observable behaviours beforehand. These behaviours can be categorised as social signals and include looking around for staff, pacing back and forth, and avoiding other customers [7]. By detecting one's social signals, it is possible to make predictions about one's future behaviour [8]. Thus, an automated approach to doing so could provide transparency, enabling us to determine how an algorithm makes a decision.

In this paper we have derived a set of 15 social signal attributes which can be used for detecting shoplifting, based on previous findings [7], [9]–[11] and our own observations. Furthermore, we have evaluated the effectiveness of these attributes for shoplifting detection by developing a novel dataset using real surveillance footage of shoplifters and genuine shoppers. The remainder of the paper is outlined as follows: in Section II we discuss the literature and the current methods; in Section III we will outline the set of attributes and the justifications for selection. Section IV outlines our experimental setup and our dataset and in Section V we will discuss these results. Finally, Section VI will conclude the paper.

II. BACKGROUND

According to official police statistics, shoplifting remains one of the most common forms of theft [12]. To combat this, retailers are spending increasing amounts on time and money on security. According to the British Retail Consortium, retailers in the United Kingdom spent over £1 billion on crime prevention in 2018; almost four times as much as was spent in 2014. Despite this, customer theft is on the rise, accounting for £663 million in losses over the same period [1].

The installation of closed-circuit television cameras (CCTV) is one commonly used security method which is often employed by retailers to deter criminals. However, research has shown that unless footage is actively monitored, surveillance cameras will prove ineffective at preventing crime [2]. Furthermore, the research conducted in [13] showed that thieves use several techniques in order to avoid detection. These included using their body to conceal theft, becoming immersed within a crowd, and wearing a disguise such as a cap or a hoodie. Without the proper training it can be difficult for those monitoring the footage to detect these behaviours and prevent theft. This is compounded by the fact that those monitoring the footage will quickly become fatigued and may miss important

 ISBN: 978-1-61208-850-1

indicators if they have to monitor several video cameras for prolonged periods [14].

The work in [4] aimed to detect suspicious behaviour by training a 3D-Convolutional neural network (3D-CNN) model. To do this, they proposed a new model for processing surveillance footage by segmenting each video into three distinct sections:

- **Strict crime movement -** The segment of the video where the individual commits the crime.
- **Comprehensive crime movement -** The precise moment when an ordinary person can detect the suspects intentions.
- **Pre-crime behaviour -** The individual's behaviour from the time they enter the store until the comprehensive crime movement begins.

They then trained their computer vision model using video footage of pre-crime behaviour in order to detect potential shoplifters. Building on this work, [15] expanded the definition of suspicious behaviour to include actions preceding other crimes, such as arson or abuse, and managed to improve the accuracy of the approach. They found that trying to find suspicious behaviour of a particular type was difficult due to the high visual similarity between suspicious and non-suspicious behaviours. Their key finding was that a binary classification approach for a generalised suspicious behaviour achieved higher accuracy than using a multi-class approach.

Both [4] and [11] use deep neural networks, trained using raw video footage taken of individuals before they shoplift. Previous research has shown that the use of these types of black box machine learning methods for this type of application can be problematic. The most obvious issue is the fact that it is very difficult to determine whether the algorithm is learning to classify potential shoplifters based on their pre-crime behaviours, or if it is learning to classify shoplifters based on some other aspect such as potential biases within the dataset [16]. The work of [17] outlines the need for transparent, interpretable machine learning approaches for high stakes learning problems such as this.

Human action recognition tasks such as shoplifting prediction can be achieved through the detection of social signals. First identified by [18], social signals are defined as "the observable behaviours displayed by an individual". Social signals can be used to infer an individual's intentions and to make predictions about their future behaviour. For example, the work of [19] used a number of vocal based social signals to determine the level of conflict within political radio debates. Social signals are generally defined in terms of five key modalities: physical appearance, vocal features, posture and gestures, facial features and interpersonal distance (Proximetrics) [8]. The automatic extraction of social signals from each of these modalities encompass a wide range of open problems within the field of pattern recognition.

In order to implement a social signal processing approach for shoplifting prediction, it is necessary to first determine a set of social signals which can accurately predict the behaviour. To this end we present a set of fifteen social signal attributes for the task of shoplifting prediction, based on the current literature, and verified through the use of a manually annotated dataset of social signal attribute taken from real shoplifting videos. This will facilitate the development of automated computer vision approaches that are interpretable (i. e. , where we can see why the model came to its decision), as well as helping to provide some clarity around the effectiveness of these attributes for the task of social signal processing. These are outlined in Section III.

III. Attribute Selection

To develop a set of social signal visual attributes for shoplifting prediction, we first examined the psychology and criminology literature in order to create an initial set of approximately 60 potential attributes. We then reduced that number by combining similar attributes and removing those which would be impossible to detect using computer vision techniques. This process results in a compact set of 15 social signal attributes as outlined below.

A. How many individuals are with them?

According to [20], most organised retail crime is committed by a group of two or more individuals. Therefore, observing whether an individual is alone or with a group can help determine if they are a potential shoplifter.

B. Are their staff members visible within the video?

According to [7], shoplifters are less likely to attempt to take items when there is a member of staff nearby, as they perceive that there is a higher risk of getting caught. Further, it has been shown that placing desirable items closer to the registers or security guards reduces the incidences of theft of those items [9].

C. What gender is the individual?

This attribute was important to determine as certain behaviours may or may not be suspicious depending on the individual's gender [21].

D. What gender is their accomplice?

Similar to C, this attribute was important to determine as certain behaviours may or may not be suspicious depending on the individual's gender.

E. Duration of time spent in the video

According to [7], individuals who are shoplifting are constantly on the lookout for security and customers or staff watching them. As a result, they may take longer to perform certain actions than a normal customer. This attribute was measured in seconds and is calculated based on the amount of time the individual is observed in the video. However, this does not necessarily correlate to the total time spent in the store; only when the individual entered and left the cameras view.

F. Are they watching staff or other customers?

The work in [7] found that individuals who are planning on shoplifting are often on the lookout for staff or other customers. Due to the difficulty in determining where an individual is looking, we defined this attribute to be true if they exhibited two or more of the following observing behaviours:

 ISBN: 978-1-61208-850-1

I. Do they clearly look around for other customers or staff before picking up an item?
II. Do they pick up an item while looking towards a member of staff?
III. Does their accomplice look out for staff or other customers while they are picking up an item?
IV. Do they frequently look towards shop staff?
V. Do they appear to be interested in the shopkeeper's interactions with other customers?

G. Do they exhibit avoidance behaviours?

In order to prevent being detected, shoplifters will often try to avoid security staff or other customers and to prevent them from seeing what they are doing. Therefore, if an individual appears to be exhibiting avoidance behaviours, such as waiting for individuals to move away from them and pacing back and forth to the same area of the shop, this may be because they are waiting for an opportunity to steal a targeted item [7]. To determine whether or not the individual was exhibiting avoidance behaviours, we used a weighted metric where one point was added for each of the following four behaviours which indicate avoidance:

I. Do they deliberately go to an area of the shop where they are not visible to the shopkeeper or security staff and stop and stay there for more than 5 seconds?
II. Do they pick up an item while the shopkeeper's back is turned to them?
III. Do they wait until other customers move away from them before picking up an item?
IV. Do they pace back and forth to a specific location before picking up an item?

Additionally, a point was subtracted if any of the following behaviours which indicate non-avoidance were observed:

I. Do they pick up an item while visible to the shopkeeper?
II. Do they pick up an item while visible to other shoppers?

If the final score for the metric was found to be one or higher, the individual was deemed to have exhibited avoidance behaviours.

H. Is the shopkeeper distracted while they pick up an item?

If an individual is contemplating shoplifting, they will wait for the shopkeeper to be distracted before attempting to hide the item. This is particularly problematic with professional thieves, where often one individual will be charged with distracting the shopkeeper while the other steal the items [20]. We defined this attribute as true if the shopkeeper was distracted while an individual picked up an item, or if either they or one of their accomplices distracted the shopkeeper (e. g. , by asking for something on a shelf behind the shopkeeper), and then picked up an item.

I. Do they appear to hide what they are doing?

If individuals are planning to steal, they may attempt to hide what they are doing from the store staff or security cameras by using either their body or an object, such as a blanket or an umbrella [9]. Therefore, it is worth noting if an individual appears to be attempting to hide themselves in this way as it may indicate that they are attempting to shoplift.

J. Do they place an item out of view either into their bag or into their pocket or else give an item to their accomplice?

Individuals who are shoplifting will all often conceal a stolen item either in a bag or in a coat before leaving. Therefore, it is important to detect if they have placed an item out of view in this way [7], [13].

K. Potential difficulty to steal the item

According to [7], one method which can be used to reduce shoplifting incidences is to place high value items closer to tills or behind the counter in order to make them more difficult to steal. Therefore, items in these locations may be more likely to be targeted by organised criminals as opposed to impulsive actors. We classified the difficulty to steal a given item on a scale of 1 to 3, where a score of 1 means the item has very little security and 3 means that the item was well guarded. This score was determined depending on whether the item was kept behind the counter, how far the item was from the entrance/exit to the store, and how likely the item was to have a security tag.

L. Are they wearing a hood baseball cap or some other clothing items that hide their appearance?

Individuals who are planning to shoplift may attempt to disguise their appearance by wearing clothing that makes it difficult to identify them from surveillance videos [13]. This may include hoodies, baseball caps, etc.

M. Are they wearing baggy clothing or carrying a bag that could potentially conceal an item?

As well as wearing clothing that may conceal their appearance, individuals may wear baggy clothing that would make it easier to conceal a stolen item, such as large coats, baggy trousers, etc. [13]. Therefore, it is worth detecting if an individual appears to be wearing this type of clothing as it may indicate that they are a potential risk.

N. Do they pick up an item and appear to be interested in it?

If an individual picks up an item and appears to examine it, this may be a sign that they are waiting for staff or other customers to move away from them before they conceal the item. Additionally, they may be examining an item for security tags [10].

O. Does the video show them interacting with staff before leaving?

According to [7], individuals who have shoplifted are likely to exit the shop quickly and try to avoid interacting with staff in case they are caught. Therefore, if individuals interact with staff before leaving this may indicate that they are not trying to avoid staff.

IV. Experiments

A. Dataset

In order to evaluate the effectiveness of these social signal attributes for the problem of shoplifting prediction, it was

 ISBN: 978-1-61208-850-1

TABLE I. OPTIMISED HYPER PARAMATERS FOUND USING A GRID SEARCH

Optimised Hyper parameters	
Model	Parameters
SVM	C=1, Gamma=0. 01, Kernel=Linear
KNN	Distance Metric = Manhattan, Neighbours=4, weights= Distance
Decision Tree	Criteria=Entropy, Max depth=5, Max features=log2, Minimum samples split=6, splitter=random
Random Forest	Criterion= gini, max depth=6, max features=log2, minimum samples split = 2, number of estimators=30
MLP	Activation Function= Identity, Hidden layers=2, layer sizes =(250,350) Solver=lbfgs

necessary to develop a novel dataset. This was done by using videos from the University of Central Florida Crimes dataset (UCF-Crimes). This dataset contains video clips for a large number of different criminal behaviours, such as arson and assault, as well as control videos. However, for these experiments we were only interested on the videos relating to shoplifting. For each video, a human observer manually annotated whether or not they observed a particular attribute as listed above. For the control group, we used the videos from the UCF-crimes dataset which were based in a retail setting and where the individual being observed made a legitimate purchase.

Attribute A was recorded as an integer denoting the number of other people with the shopper or shoplifter. Attributes C and D were encoded using one hot encoding in order to prevent the model from inferring some ordered relationship. Therefore, for attribute C there was two values "Gender is Male" and "Gender is Female," and for attribute D there was two attributes "Accomplice is Male," and "Accomplice is Female." If there was more than one accomplice, the gender was encoded as the majority of the group. Attribute E was recorded as an integer denoting the amount of time the individual was observed in seconds. Attribute K was recorded as an integer value between 1 and 3 as detailed in section III. The remaining attributes were all recorded as either true or false. This resulting dataset contained a total of 93 records, with 48 shoplifting records and 45 control records.

B. Experimental design

In order to evaluate the effectiveness of these attributes, we used them to train a diverse set of supervised learning models and evaluated them in terms of accuracy, precision, and recall. These models were as follows: Support vector classifier (SVM) [22], K-Nearest neighbours (KNN) [23], Classification and regression decision tree (CART) [24], Random Forest [25] and multi-layer perceptron (MLP) [26]. Each model was evaluated using five-fold cross validation and the hyper parameters were optimised using a grid search.

For the support vector classifier, we used a grid search to find the optimal kernel, and the optimal values for C and Gamma. The kernels used in the grid search were linear, radial basis function, polynomial and sigmoid. The values for the regularization parameter C used in the grid search were: 1,10,100 and 1000. Finally, the values used for gamma in the grid search were: 10^2, 10^3 10^4.

For the KNN classifier, we used a grid search to find the optimal distance function, optimal number of neighbours, and if the weights of the neighbours were uniform or weighted based on distance. The distance metrics used in the grid search were Euclidian distance, Manhattan distance, Chebyshev distance, and Minkowski distance. For K we tested the range of values between 1 and 16.

For the decision tree classifier, we used a grid search to determine: the optimal criterion used to measure the quality of the split, either using gini impurity or entropy; the optimal strategy used to split each node, either using the feature with the highest importance or using a random feature, with the random distribution weighted by importance; the max tree depth, between None up to 8; the minimum samples needed to split a node again from None up to 8, and finally, the number of features to consider when looking for the best split, either the square root of the number of features, $\log_2$ of the number of features, or just using the entire set of features.

For the random forest classifier, we used a grid search to determine the optimal values for the criterion, max depth, minimum number of samples needed to split a node, and the number of features to consider when looking for the best split. We also optimised for the number of trees used in the forest in multiples of 5 from 25 up to 100.

Finally, for the MLP model, we used a grid search to find: the optimal activation function, either using an identity function (where $F(x) = x$), a logistic activation function, a hyperbolic tan activation function or a rectified linear activation function; the optimiser, either stochastic gradient decent, an Adam optimiser or the LM-BFGS optimiser; and the optimal number of neurons for each of the two hidden layers, in increments of 50 from 50 up to 400. The optimised parameters for each of these models are presented in Table I.

V. RESULTS

We evaluated the performance of each of our learning models in terms of accuracy, precision, and recall, where accuracy is calculated as:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad (1)$$

Precision is calculated as:

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

And recall is calculated as:

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

where TP is the number of true positive predictions, FP is the number of false positive predictions, TN is the number of true

TABLE II. RESULTS FROM MACHINE LEARNING METHODS

Method	Accuracy	Precision	Recall
SVM	92.40%	93.05%	92.44%
KNN	80.64%	82.09%	81.00%
Decision Tree	83.92%	84.36%	86.78%
Random forest	94.50%	93.75%	91.33%
MLP	92.40%	91.85%	92.44%

negative predictions and FN is the number of false negative predictions. The results for each approach are presented above.

As can be seen in Table II, the Random Forest approach was found to be the most accurate on our dataset, with an accuracy of 94. 5%. This was followed by the SVM approach, the MLP approach the decision tree approach and the KNN approach. This was the same for the precision metric. However, for the recall metric, both the SVM and MLP approach slightly outperformed the random forest approach. This is important as the recall metric determines how many of the individuals who were genuinely shoplifters were detected as such. However, if the system fails to highlight a suspicious individual who does go on to shoplift, then that individual may evade detection.

These results indicate that there are clear and measurable differences between the social signals exhibited by shoplifters and those exhibited by regular shoppers.

Figure 1 shows the feature importance generated by the random forest model [27]. The results here indicate that the most significant attribute was: "Do they exhibit avoidance behaviours." This was followed by: "Do they interact with staff before leaving," "potential difficulty to steal the item," "Do they place the item out of view," and "Do they appear to hide what they are doing." These attributes almost all relate to the individual performing (or not performing) a given action, which may indicate that an individual's behaviour gives a more reliable indicator of their intention than environmental factors, such as their clothing. Conversely, the least important features were "Gender," "Gender of accomplice," "Are they wearing clothing items that could hide their appearance," and "Are their staff members visible withing the shot." Again, this makes sense, as these are all environmental attributes which don't give an indication about how an individual is behaving.

As well as generating the feature importance, we also performed sensitivity analysis on each of the attributes. This was done by removing each attribute individually and then evaluating the change in performance. As can be seen from the

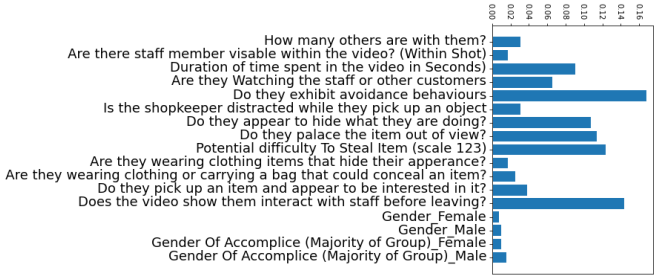


Figure 1. Feature importance from the random forest model.

TABLE III. RESULTS FROM SENSITIVITY ANALYSIS

Attribute	Accuracy	Precision	Recall
A	-1.05%	-1.00%	-1.00%
B	-2.11%	-1.91%	-2.11%
C	-2.05%	-1.31%	-2.33%
D	0.06%	0.03%	0.11%
E	-1.11%	-0.82%	-1.11%
F	-3.22%	-3.02%	-3.22%
G	-4.27%	-3.93%	-4.33%
H	-2.11%	-1.91%	-2.11%
I	-3.16%	-1.96%	-3.33%
J	-2.11%	-1.91%	-2.11%
K	-1.05%	-1.00%	-1.00%
L	-2.05%	-1.59%	-2.11%
M	-2.16%	-1.82%	-2.11%
N	1.11%	1.00%	1.11%
O	-5.26%	-5.13%	-5.33%

results in Table III, the most important attributes are: F *"are they watching staff or other customers,"* G *"Are they exhibiting avoidance behaviours,"* I *"Do they appear to hide what they are doing,"* and O *"Do they interact with staff before leaving*."

This is consistent with the results found by calculating the feature importance. As discussed above, these attributes all relate to the individual performing a specific action, which may be why they are stronger features for predicting shoplifting. It is interesting to note that removing attribute N "*Do they pick up an item and appear interested in it"* caused the model to improve in accuracy. This may indicate that this attribute is a poor indicator of potential shoplifting or that it may appear too frequently in both groups to be useful. All of the other attributes showed some decrease in classification accuracy when removed.

VI. CONCLUSION

In this paper we have outlined the need for a transparent interpretable model for the problem of suspicious behaviour detection. To this end we developed a set of 15 social signal visual attributes which have been used to predict if an individual is likely to shoplift. We demonstrated the effectiveness of these attributes for the problem of shoplifting detection by manually annotating a subset of videos from the UCF-crimes dataset. We evaluated the effectiveness of each attribute by calculating the feature importance and through the use of sensitivity analysis. These results showed that detection of attributes which show individuals performing actions, such as interacting with staff and exhibiting avoidance behaviours, were the strongest indicators of whether or not an individual was a potential shoplifter.

For future work, these results will be validated using a larger dataset of shoplifting videos. Currently, the UCF-Crimes dataset [28] is the largest open-source dataset for this problem.

 ISBN: 978-1-61208-850-1

However, this dataset only contains 50 videos of shoplifting; a number of which are cut short and don't provide enough footage to definitively determine the social signals exhibited before a shoplifting attempt. Furthermore, the videos in this dataset come from a number of different retail environments. A single dataset of retail shoplifting from a single store with multiple cameras and which contains both genuine customers and thieves, and where each customers entire history, from the moment they enter the store to the moment they leave, would enable us to determine more definitively which social signals are suspicious, and the frequency at which they occur.

Secondly, we suggest that there may be other social signals that indicate that an individual is likely to shoplift which we may have missed, or which may not be present in the current literature. Further to this, it has already been noted that individuals who are shoplifting have developed techniques to attempt to evade security measured. It is inevitable then that once these methods are implemented individuals will develop new techniques in order to evade them.

REFERENCES

[1] BRC, "2019 Retail Crime Survey," United Kingdom, 2019.

[2] A. Spriggs and M. Gill, "CCTV and Fight Against Retail Crime: Lessons from a National Evaluation in the U. K. ," *Secur. J.* , vol. 19, no. 4, pp. 241–251, 2006, doi: 10. 1057/palgrave. sj. 8350023.

[3] T. A. Smith, "Investigations: Consumer Retail Shoplifting," *Encycl. Secur. Emerg. Manag.* , pp. 1–7, 2020, doi: 10. 1007/978-3-319-69891-5_172-1.

[4] G. A. Martínez-Mascorro, J. C. Ortiz-Bayliss, J. R. Abreu-Pederzini, and H. Terashima-Marín, "Suspicious behavior detection on shoplifting cases for crime prevention by using 3D convolutional neural networks. ," *arXiv*, Apr. 2020.

[5] E. Tartaglione and M. Grangetto, "A non-discriminatory approach to ethical deep learning," *arXiv*. 2020, doi: 10. 3390/computation9020024.

[6] A. Završnik, "Criminal justice, artificial intelligence systems, and human rights," *ERA Forum*, vol. 20, no. 4, pp. 567–583, 2020, doi: 10. 1007/s12027-020-00602-0.

[7] C. Cardone and R. Hayes, "Shoplifter Perceptions of Store Environments: An Analysis of how Physical Cues in the Retail Interior Shape Shoplifter Behavior," *J. Appl. Secur. Res.* , vol. 7, no. 1, pp. 22–58, 2012, doi: 10. 1080/19361610. 2012. 631178.

[8] J. J. K. Burgoon, N. Magnenat-Thalmann, M. Pantic, and A. Vinciarelli, *Social signal processing*, First Edit. Cambridge University Press, 2017.

[9] N. V. Lasky, B. S. Fisher, and S. Jacques, "'Thinking thief' in the crime prevention arms race: Lessons learned from shoplifters oa," *Secur. J.* , vol. 30, no. 3, pp. 772–792, 2017, doi: 10. 1057/sj. 2015. 21.

[10] D. A. Dabney, R. C. Hollinger, and L. Dugan, "Who actually steals? A study of covertly observed shoplifters," *Justice Q.* , vol. 21, no. 4, pp. 693–728, 2004, doi: 10. 1080/07418820400095961.

[11] P. W. Fombelle *et al.* , "Customer deviance: A framework, prevention strategies, and opportunities for future research," *J. Bus. Res.* , vol. 116, no. November 2019, pp. 387–400, 2020, doi: 10. 1016/j. jbusres. 2019. 09. 012.

[12] C. S. Branch, "Trends in Police Recorded Crime in Northern Ireland," Northern Ireland, 2020.

[13] M. Gill, *Shoplifters on Shop Theft: Implications for Retailers, Perpetuity Research & Consultancy International*, First edit. Leicester: Perpetuity Research and Consultancy International (PRCI), 2007.

[14] R. Clarke and G. Petrossian, *Shoplifting Problem-Oriented Guides for Police Problem-Specific Guides Series*. 2018.

[15] G. A. Martinez-Mascorro, J. Carlos Ortiz-Bayliss, and H. Terashima-Marin, "Detecting Suspicious Behavior on Surveillance Videos: Dealing with Visual Behavior Similarity between Bystanders and Offenders," *IEEE ANDESCON*, pp. 1–7, 2020.

[16] C. Sandvig, K. Hamilton, K. Karahalios, and C. Langbort, "When the algorithm itself is a racist: Diagnosing ethical harm in the basic Components of Software," *Int. J. Commun.* , vol. 10, no. June, pp. 4972–4990, 2016.

[17] C. Rudin, "Stop Explaining Black Box Machine Learning Models for High Stakes Decisions and Use Interpretable Models Instead," *Nat. Mach. Intell.* , vol. 1, pp. 206–215, 2019.

[18] A. Vinciarelli, M. Pantic, and H. Bourlard, "Social signal processing: Survey of an emerging domain," *Image Vis. Comput.* , vol. 27, no. 12, pp. 1743–1759, 2009, doi: 10. 1016/j. imavis. 2008. 11. 007.

[19] S. Kim, F. Valente, and A. Vinciarelli, "Automatic detection of conflicts in spoken conversations: Ratings and analysis of broadcast political debates," in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, 2012, pp. 5089–5092, doi: 10. 1109/ICASSP. 2012. 6289065.

[20] K. M. Finklea, "Organized retail crime," *J. Curr. Issues Crime, Law Law Enforc.* , vol. 5, no. 3, pp. 163–187, 2012.

[21] H. Hirtenlehner and K. Treiber, "Can Situational Action Theory Explain the Gender Gap in Adolescent Shoplifting? Results From Austria," *Int. Crim. Justice Rev.* , vol. 27, no. 3, pp. 165–187, 2017, doi: 10. 1177/1057567717690199.

[22] C. CORTES and V. VAPNIK, "Support-Vector Networks," *Mach. Learn.* , vol. 7, no. 5, pp. 63–72, 1992, doi: 10. 1109/64. 163674.

[23] N. S. Altman, "An introduction to kernel and nearest-neighbor nonparametric regression," *Am. Stat.* , vol. 46, no. 3, pp. 175–185, 1992, doi: 10. 1080/00031305. 1992. 10475879.

[24] L. Breiman, J. Friedman, C. J. Stone, and O. Richard A. , *Classification and regression trees*. CRC Press, 1984.

[25] L. Breiman, "Random forests," *Mach. Learn.* , vol. 45, no. 1, pp. 5–32, Oct. 2001, doi: 10. 1023/A:1010933404324.

[26] M. L. Minsky and S. A. . Papert, *Perceptrons*, Expanded. The MIT Press, 1988.

[27] G. Louppe, "Understanding Random Forests: From Theory to Practice," 2014, [Online]. Available: http://arxiv. org/abs/1407. 7502.

[28] W. Sultani, C. Chen, and M. Shah, "Real-World Anomaly Detection in Surveillance Videos," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.* , pp. 6479–6488, 2018, doi: 10. 1109/CVPR. 2018. 00678.

 ISBN: 978-1-61208-850-1

Recovering Shape from Endoscope Image Using Eikonal Equation

Yuji Iwahori
Dept. of Computer Science
Chubu University
Kasugai, 487-8501 Japan
Email: iwahori@isc.chubu.ac.jp

Hiroyasu Usami
Dept. of Computer Science
Chubu University
Kasugai, 487-8501 Japan
Email: usami@isc.chubu..ac.jp

M. K. Bhuyan
Dept. of Electronics and Electrical Eng.
Indian Institute of Technology Guwahati
Guwahati, 781039 India
Email: mkb@iitg.ac.in

Aili Wang
Higher Education Key Lab.
Harbin University of Sci. and Tech.
Harbin, China 150006
Email: aili925@hrbust.edu.cn

Naotaka Ogasawara
Dept. of Gastroenterology
Aichi Medical University
Nagakute, 480-1195 Japan
Email: nogasa@aichi-med-u.ac.jp

Kunio Kasugai
Dept. of Gastroenterology
Aichi Medical University
Nagakute, 480-1195 Japan
Email: kuku3487@aichi-med-u.ac.jp

***Abstract*—This paper proposed a shape recovery approach from Endoscope Image using Eikonal Equation. Photometric constraint equation derived from the Lambert reflectance and geometrical constraint equation derived from the relationship between the neighboring points are used and these equations can make a new approximation equation of Eikonal equation under the point light source illumination and perspective projection. The original endoscope image is transformed and generated to the Lambertian image by removing the specular reflectance. Framework of Fast Marching Method using the derived Eikonal Equation can recover the 3D shape from endoscope image. Usefulness was confirmed using simulation and experiments.**

Keyword: Shape, Endoscope, Eikonal Equation, Fast Marching Method

I. INTRODUCTION

Endoscope is used in the medical diagnosis to detect polyps and the examinations are performed for the purpose of finding abnormal parts, such as bleeding, inflammation in the internal organs of the stomach and intestines (See Refs.[1][2]). Polyps found by endoscope have a variety of sizes and shapes. This paper proposed a new approach for a polyp shape and size recovery by solving Eikonal Equation under the condition of point light source illumination and perspective projection. Shape is recovered using an extended Fast Marching Method (FMM) approach. Proposed method is described in Section II. In Section III, experimental results are shown with absolute size and corresponding polyp shape. The approach provides overall good performance for the supporting system of medical diagnosis. Finally Conclusion is provided in Section IV.

II. PROPOSED METHOD

Endoscope image is obtained under the white light source. Procedure of the proposed approach is shown as follows.

A. Removal of Specular Reflectance Component and Generation of Lambertian Image

Our previous approach proposed in Ref.[3] is applied to remove the specular reflectance component and generation of Lambertian image with uniform surface reflectance from the original endoscope image. Converting the RGB (Red-Blue-Blue) to HSV (Hue-Saturation-Value) representation, classification for reflectance using the H histogram, then uniform reflectance image processing is performed using the ratio of V based on the difference of reflectance. Procedures are as follows.

Step1. Classification using histogram of H
Step2. Calculate the V ratio between interest color points and those neighboring points whose color is most frequent color.
Step3. Equalization of reflectance using V ratio calculated in Step2 and using the points, which are not used in Step2 of interest color.
Step4. Equalization of reflectance for all color groups by repeat Step2 and Step3.

B. 3D Shape Recovery

Photometric constraint equation and geometrical constraint equation derived should become the same depth value Z for both (See Ref.[4]). These equations can make a new approximation equation of Eikonal equation under the point light source illumination and perspective projection.

$$\sqrt{\frac{CV(-px-qy+f)}{E(p^2+q^2+1)^{\frac{1}{2}}}} = \frac{Z_k(f-p_kx_k-q_ky_k)}{f-p_kx_t-q_ky_t} \quad (1)$$

Here, $t(trial)$ represents a trial point and $k(known)$ represents a known point of depth Z. C represents the reflectance

 ISBN: 978-1-61208-850-1

factor, (p, q) represent gradient parameters, (x, y) represent image coordinate, f represents a focal length of the lens, E represents the observed image intensity and $V = f^2/(x^2 + y^2 + f^2)^{\frac{3}{2}}$.

Expanding Eq.(1) gives

$$\sqrt{p^2+q^2+1} = \frac{CV(f-p_kx-q_ky)^2(f-px-qy)}{EZ_k^2(f-p_kx_k-q_ky_k)^2} \quad (2)$$

Approximation of $(f - p_kx - q_ky) \fallingdotseq (f - px - qy)$ derives

$$p^2+q^2+1 = \frac{C^2V^2(f-p_kx-q_ky)^6}{E^2Z_k^4(f-p_kx_k-q_ky_k)^4} \quad (3)$$

Then the following equation is derived.

$$\sqrt{p^2+q^2} = \sqrt{\frac{C^2V^2(f-p_kx-q_ky)^6}{E^2Z_k^4(f-p_kx_k-q_ky_k)^4} - 1} \quad (4)$$

Depth Z can be obtained by FMM algorithm developed in Ref.[5]. Here, it is possible to estimate the value of reflectance parameter C using two images and feature points matching to estimate the absolute size of polyp since absolute size is important and it depends on the value of reflectance factor C.

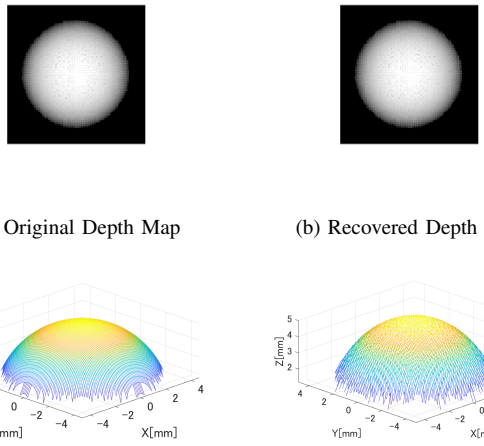
Step1. SIFT (in Ref.[6]) or ORB (in Ref.[7]) feature points extracted from blood vessels using two images and movement ΔZ of endoscope camera is estimated.

Step2. Parameter C is estimated using ΔZ (See Ref.[6]).

Step3. Shape recovery is applied for each uniform Lambertian image generated by Ref.[3].

III. Experiment

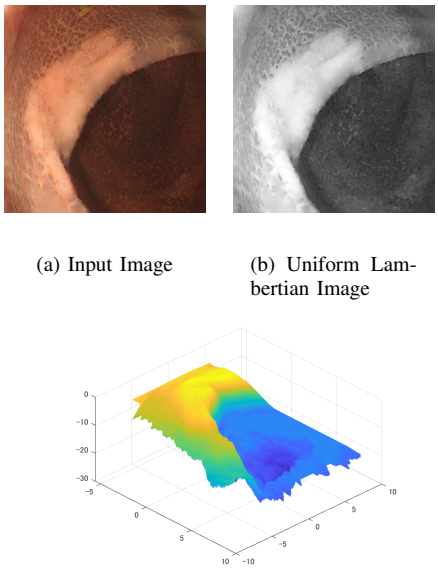
Simulation image for a hemisphere whose center is $(0, 0, 15)$ with the radius $R = 5$[mm], focal length $f = 10$[mm], reflectance factor $C = 22950$ is assumed and the result is shown in Figure 1. 3D shape are recovered for endoscope image as shown in Figure 2.



(a) Original Depth Map (b) Recovered Depth Map

(c) Original Shape (d) Recovered Shape

Fig. 1. Computer Simulation



(a) Input Image (b) Uniform Lambertian Image

(c) 3D Shape

Fig. 2. Recovered 3D Shape

IV. Conclusion

This paper proposed a method to recover 3D shape and size from endoscope image using Eikonal Equation. Absolute size recovered from estimated C becomes also important from two images under endoscope motion. Further subject includes to use the recovered shape to the undetected or misclassification of polyp for the support of medical diagnosis.

Acknowledgment

Iwahori's research is supported by JSPS Grant-in-Aid Scientific Research (C)(#20K11873).

References

[1] S. Hwang, J. Oh, W. Tavanapong, J. Wong, and P. C. D Groen, "Polyp detection in colonoscopy video using elliptical shape feature", *IEEE ICIP 2007*, Vol. 2, pp. II–465, 2007.

[2] T. Ooto et al., "Cost Reduction of Creating Likelihood Map for Automatic Polyp Detection Using Image Pyramid", *ACIT/CSII/BCD 2017*, pp.204-209, 2017.

[3] N. Ikeda, H. Usami, Y. Iwahori, B. Kijsirikul, and K. Kasugai, "Generating Lambertian Image by Removing Specular Reflection Component and Difference of Reflectance Factor Using HSV", Proc. of ITC-CSCC 2016, pp.547-550, 2016.

[4] K. Tatematsu et al., "Shape from Endoscope Image based on Photometric and Geometric Constraints", Procedia Computer Science, Elsevier, Vol.22, pp.1285-1293, 2013.

[5] R. Kimmel and J. A. Sethian : "Optimal Algorithm for Shape from Shading and Path Planning", *JMIV*, Vol. 14, pp. 237-244, 2001.

[6] Y. Iwahori et al., "Estimating Reflectance Parameter of Polyp Using Medical Suture Information in Endoscope Image", ICPRAM 2016, pp.503-509, 2016.

[7] H. Toda et al., "Shape Recovery of Polyp using Blood Vessel Detection and Matching Estimation by U-Net", IIAI AAI 2019, pp. 450-453, 2019.

 ISBN: 978-1-61208-850-1

Detection of Gas Flares Using Satellite Imagery

Alexander Troussov, Dmitry Botvich and Sergey Vinogradov
International laboratory for mathematical modelling of social networks,
RANEPA,
Moscow, Russia
e-mails: troussov@gmail.com, dbotvich@gmail.com, derbosebar@gmail.com

***Abstract*—During the extraction, transportation and processing of oil, associated petroleum gas is formed, which is usually disposed in flares. Monitoring these flares is an important environmental challenge. Objective instrumental methods for detecting gas flares and assessing the volumes of gas burnt on them are based on multi-spectral remote sensing of the nighttime Earth. To refine the list, a manual check is used, which includes a visual examination of the locations of the alleged gas flares on high-resolution daytime images. Automating this check reduces the cost of monitoring flares. The paper proposes a method for verifying the list of high-temperature anomalies, based on the classification of images of objects in daytime images. The classification is carried out using machine learning methods.**

Keywords*—night lights; light pollution; viirs; image processing; machine learning; deep learning.

I. INTRODUCTION

During the extraction, transportation and processing of oil, associated petroleum gas is formed, which is usually disposed in flares. Monitoring these flares is an important environmental challenge (see materials of the World Bank's Global Gas Flaring Reduction Partnership). Objective instrumental methods for detecting gas flares and assessing the volumes of gas burnt on them are based on multi-spectral remote sensing of the nighttime Earth. The sensor Visible Infrared Imaging Radiometer Suite (VIIRS) is used by satellites (Suomi National Polar-orbiting Partnership (Suomi NPP) and NOAA-20 weather satellite) at nighttime to collect imagery and radiometric measurements of the land, atmosphere and oceans in the visible and infrared bands of the electromagnetic spectrum. These data make it possible to find high-temperature anomalies on the Earth's surface, and by the spectrum of radiation to distinguish gas flares from, for example, forest fires and greenhouses. The most famous algorithm for finding gas flares from Earth remote sensing data is the VIIRS Nightfire (VNF) algorithm [1], [2], [3].

The refinement of this algorithm for use on the territory of Russia is described in our paper [4]. However, the list of gas flares obtained using the VNF method is not completely correct. Among the main factors that make the presence of errors inevitable, we mention the low temporal resolution of satellite data (satellites do not often fly over objects, objects can be covered by clouds), and interference in observations that are difficult to eliminate (snow cover, polar nights). Additional work is required to compile a "final list" of gas flares from a "preliminary list" of gas flares. In the "final list" it is also desirable to classify the flares by type of enterprise (upstream, transportation, downstream processing).

In our paper [4], methods of manual correction of the list of gas flares for the territory of Russia are described based on the use of various additional sources and databases. For example, the presence of an object in the authoritative list CEDIGAS (The International Association For Natural Gas), is a confirmation of the eligibility of including the object in the "final list". If the coordinates of an object are not included in the boundaries of licensed areas for oil production, then in the conditions of Russia this object can be confidently excluded from objects of the upstream type. However, such manual corrections are laborious and time-consuming, and flare monitoring needs to be done on a continuous basis with a high temporal resolution. Automation of verification, even partial, reduces the cost of monitoring flares. If we consider the monitoring of gas flares in areas of armed conflict, where illegal oil production and processing may take place, then the automation of gas flare testing becomes uncontested.

In this paper, we propose a method for checking the list of high-temperature anomalies obtained on the basis of NTL data, based on the classification of images of objects in daytime images. The classification is carried out using machine learning methods.

The paper is organized as follows. In Section II, we describe the satellite data used in the paper. Our approach to the correction of gas flares list is outlined in Section III. In Section IV, we present the image transformation process, including image coding, image descriptor extraction, etc. In Section V, we present some preliminary results. Finally, Section VI concludes the paper.

II. SATELLITE IMAGE DATA

The original flare list used in this work is part of the list of associated petroleum gas flares in Russia obtained in the course of our work described in [4]. For each prospective gas flare, a high-resolution daytime image was selected that covered the location of the supposed flare. Some of these data are labeled - that is, it is known whether or not there is a gas flare a given point. In addition, during the training phase, we added a manually selected set of daytime images that do not contain gas flares, but there are production facilities that may visually resemble oil and gas production and processing facilities.

 ISBN: 978-1-61208-850-1

The VNF algorithm utilizes near-infrared and short-wave infrared data at night, gathered by the VIIRS on board satellites NPP and NOAA-20 [3]. In [4] we used a version of the VNF algorithm adapted to the conditions of Russia. The total number of gas flares detected in [4] is about two and a half thousand.

We used also daytime satellite imagery from the webserver TerraServer to compile a “final list” of gas flares based on a “tentative list” of gas flares. The resolution of the daytime images is about 1m per pixel.

III. CHECKING AND CORRECTION OF GAS FLARES LIST

First of all, we note that a wide variety of third-party external sources can also be used when considering whether a snapshot contains a gas flare. For example, in many countries the licensing of oil areas for oil production (for example, in the USA, Saudi Arabia, Russia, etc.) works quite effectively(and reliably. The lists of such areas (i.e., their coordinates) are publicly available and can be used for the evaluation if a particular area is actually the oil area.

Here, we consider a different approach based on the daytime satellite image classification. We are using machine learning techniques here to adjust the ”final list” of gas flares based on some ”tentative list” of gas flares. The input data for us are the geographic coordinates of the alleged gas flares, which are used to extract recent daily satellite data, and then these images are analyzed by our automatic classifier (called, ClassGasFlare), which determines the type of gas flare from the image (by specialization enterprises: production, transportation, upstream processing, downstream processing) or lack of flares in the picture.

Since manual correction of this data is quite laborious and time-consuming, automating the check of the list can significantly reduce this work and helps to close the entire verification technological cycle: from automatic acquisition of images from the data cloud to their classification according to the type of gas flare or the absence of gas flares. In those cases when the automatic classifier has a ”difficult” case, then only these ”difficult” cases can be left for manual additional verification, which significantly reduces the volume of manual work.

We train our classifier on a sufficiently large volume of historical images, where they are labeled as (flare-dobycha, flare-transport, flare-upstream, flare-downstream, flare-no) depending on the type of oil production or simply as (flare-yes, flare-no) to classify the presence or absence of oil production. Thus, formally, in terms of machine learning, we get the picture classification problem. In a first approximation, this is true, but there are some important aspects that significantly complicate this task. For example, aspects such as:

- images can cover partially our objects (where the gas flare is located),
- images can be ”naturally spoiled” (for example, by clouds that cover part of the surface in the image),
- images may have additional features such as snow cover, which also distorts the original.

All these factors significantly affect the quality of the classification and should be taken into account when developing a classifier.

IV. DATA FOR THE CLASSIFIERS TRAINING

This section describes the data and machine learning classifiers used for the training and evaluation of the satellite image classification. The image transformation techniques are also discussed in this section.

A. Training data sets

We use satellite imagery (from the TerraServer web site) for the period 2010-2017 (GasFlareData dataset). The resolution of images from this dataset is within 1 m per pixel. The images include both different types of gas flares and images where there are no gas flares. As a rule, pictures are presented for different moments in time (usually 3-10 variants), both in color and in black and white. Images also often contain ”clouds” and ”snow cover” as natural disturbances in our classification task. There are also in small numbers simply spoiled pictures. Figures 1, 2 and 3 show examples of images from the used dataset.

Figure 1: Examples of ”clean” upstream flare images.

Moreover, it would be more correct to consider our task as a task for the presence of certain types of scenes (scene detection), which in itself, as a rule, is much more complicated than the usual classification. We only note that we do not consider

Figure 2: Examples of ”clean” downstream flare images.

Figure 3: ”Unclean” examples of flare images: downstream flare, covered with clouds (on the left) and upstream flare, covered with snow (on the right).

the approach from the point of view of scene detection in this work.

B. Image extractors and descriptors: SIFT and SURF algorithms

This section provides an overview of the various experiments we use to evaluate the performance of the algorithm for image classification. The set of raw data used in our experiments is described here. The following describes methods for transforming images and methods for obtaining feature descriptors (including SIFT, SURF, etc.). Then, preliminary results of our work on evaluating the accuracy of the algorithm for classifying images with flares are presented.

Both of these methods have an important feature: they are relatively insensitive to both scaling and rotation of images. This is very important for our task, since our images can be with different scaling and resolution, and are oriented in different ways. The scale-invariant feature transform (SIFT) [5] [6] was the first algorithm with similar properties, and the speeded up robust features (SURF) algorithm [7] improves it overall: SURF is faster and more robust than SIFT. The result of SIFT and SURF is a certain finite-dimensional vector of descriptors, and the order of the elements itself is not important in this vector of descriptors, i.e., this means that in fact it is a set. An image descriptor vector is a kind of histogram of the visual elements it contains.

After using SIFT or SURF algorithms, one can use the corresponding descriptors of pictures instead of the pictures themselves, both at the stage of training and at the stage of using the classifier.

C. Image Data Set Augmentation

The quality and quantity of datasets is very important as this affects the accuracy of machine learning algorithms, as well as it is related to overfitting and underfitting problems. We increase the size of the training data by using augmentation techniques, e.g., cropping, rotating, shifting and scaling of original images. As SIFT and SURF are invariant with respect to rotating, shifting and scaling we do not use them with SIFT and SURF algorithms but we use cropping technique. On other hand, for deep learning models, in particular, for we use all of them: including rotating, shifting, scaling as well as cropping extensively. Overall, the image augmentation helps to improve the prediction accuracy of the model. In our experiments after applying the augmentation technique, the data set (about 11000 images) is increased approximately by factor 2 (about 22000 images).

D. Classifier training process

The encoded image feature vectors from each category of images are used in the classifier training process for the SVM and Random Forrest algorithms to create a predictive function of the model. Figure 4 demonstrates the steps of the image preparation used in our image classification approach.

 ISBN: 978-1-61208-850-1

It is based on a more generic framework for image data transformation discussed in [10].

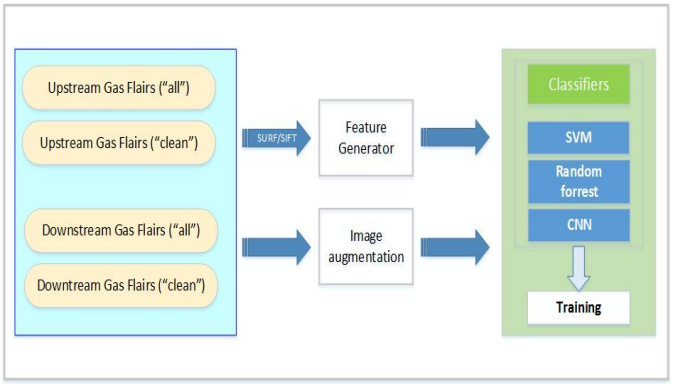


Figure 4: Stages of processing for gas flare images.

V. EXPERIMENTS WITH THE FLARE IMAGE CLASSIFICATION

In this section, we describe the experiments we use to evaluate the quality of classifiers for classifying gas flare images.

We are interested in the average accuracy measurement and the "the confusion matrix" for various experiments. For this we use different categories of images and their groups from the GasFlareData data set.

Note that the results below are *preliminary*, and the classifiers themselves are far from optimal. This is especially true for the results on the application of the deep learning (CNN classifier).

Experiment 1. Measuring classification accuracy from data ("all" or "clean") into categories (upstream and downstream) using SURF extractor for SVM [8], Random Forrest [9] and without using SURF in the case of the CNN classifier. Here a classifier (SVM, Random Forrest and CNN) is used to create a prediction model based on two image classes, including (1) "presence of a gas flare" and (2) "no gas flare". Each category of data sets (upstream and downstream) is considered separately here. In the training process, we use 70% of the entire set of images from each category for the training purposes. The rest of each image set (i.e., 30% of all images) are used during the forecast evaluation stage. Our measurements show that the achieved average forecast accuracy is about 75% (see TABLE I).

Experiment 2. Measuring classification accuracy from data ("all" or "clean") combined categories (upstream + downstream) using SURF extractor for SVM, Random Forrest and without using SURF for CNN. Here the SVM, Random Forrest and CNN classifier are used to generate a prediction model based on three image classes: ((1) "downstreanm gas flare presence", (2) "upstreanm gas flare presence" or (3) "gas flare absence"). During training we use 70% of the whole set of images (downstream + upstream) for data ("all" or "clean"). The rest of the images (i.e., 30% of all images) are included in the test data set and are used during the forecast evaluation stage. Preliminary experiments also demonstrate that the achieved average forecast accuracy for the three classes is about 70% (see TABLE II).

TABLE I. THE AVERAGE ACCURACY RESULTS FOR THE TWO CLASSES

Data	SVM	Random Forrest	CNN
Upstream "Clean"	0.81	0.79	0.75
Downstream "Clean"	0.78	0.78	0.74
Upstream "all"	0.79	0.76	0.75
Downstream "all"	0.77	0.76	0.74

TABLE II. THE AVERAGE ACCURACY RESULTS FOR THE THREE CLASSES

Data	SVM	Random Forrest	CNN
Upstream, "clean" + Downstream, "clean"	0.72	0.71	0.70
Upstream, "all + Downstream, "all"	0.70	0.70	0.68

VI. CONCLUSION AND FUTURE WORK

In this paper, a method for checking and correcting the list of high-temperature anomalies obtained on the basis of nighttime light data is proposed. The image classification of daytime satellite images is used for this purpose. It is carried out using machine learning methods.

Various machine learning methods and algorithms to classify the gas flare images from the GasFlareData set is applied. The up-to-date solutions that generally demonstrate the most acceptable classification quality in a wide variety of applications are used. The process of preparing images for classification, including special image encoding operations such as the well-known SIFT and SURF algorithms, is presented. For different variants of datasets: "clean" (including only those that do not contain clouds or snow) and "all" (including those that do not contain clouds and/or snow), the flare image classification is performed.

In experiments, the preliminary comparison of the classification quality for different machine learning methods (including SVM, random forest, Convolutional Neural Networks (CNN)) both with and without image encoding operations is carried out. Both the variants of the input data ("clean" or "all") are evaluated. The following average forecast accuracy is achieved in the experiments: about 75% for the two classes and about 70% for the three classes.

Future work concerns the improvement of the classification accuracy of the machine learning models used in the paper. We also plan to apply the other deep learning architectures to

 ISBN: 978-1-61208-850-1

the problem of interest and compare them with the all other classifiers.

ACKNOWLEDGMENT

We thank Mikhail Zhizhin at the Colorado School of Mines, Golden, USA and Alexey Poyda at the National Research Centre ”Kurchatov Institute”, Moscow, Russia for fruitful discussions and collaboration in collecting samples of satellite images.

REFERENCES

[1] Earth Observation Group, Payne Institute, Colorado School of Mines, Golden, USA [Online]. Available from: https://payneinstitute.mines.edu/eog/viirs-nightfire-vnf/ [retrieved: March, 2021].

[2] C.Elvidge, M. Zhizhin, F-C. Hsu and K. Baugh, “VIIRS nightfire: Satellite pyrometry at night”. Remote Sensing, 5, no. 9, 2013, pp.4423-4449.

[3] C. Elvidge, M. Zhizhin, K. Baugh, F-C. Hsu and T. Ghosh, “Methods for Global Survey of Natural Gas Flaring from Visible Infrared Imaging Radiometer Suite Data”. Energies, 9(1), 14, 2016, pp.1-15.

[4] A. Matveev, A. Andreev, M. Zhizhin and A. Troussov, “Satellite Monitoring of Associated Gas Flaring Torches in Russia” (In Russian). SSRN Electronic Journal, DOI: 10.2139/ssrn.3362303, January 2019.

[5] D. G. Lowe, “Object recognition from local scale-invariant features”. Proceedings of the International Conference on Computer Vision. 2. 1999, pp. 1150–1157.

[6] D. G. Lowe, “Distinctive Image Features from Scale-Invariant Keypoints”. International Journal of Computer Vision. 60 (2), 2004, pp.91–110.

[7] H. Bay, T. Tuytelaars and L. Van Gool, “Surf: Speeded up robust features”. European conference on computer vision, 2006, pp.404-417.

[8] H. Drucker, C. Burges, L. Kaufman, A. J. Smola, and V. Vapnik, “Support Vector Regression Machines”, in Advances in Neural Information Processing Systems 9, NIPS 1996, 1997, pp.155–161.

[9] L. Breiman, “Random Forests”. Machine Learning, 45, pp.5-32, 2001.

[10] S. Loussaief and A. Abdelkrim, “Machine Learning framework for image classification”. Advances in Science, Technology and Engineering Systems Journal, 3, 2018, pp.1-10.

 ISBN: 978-1-61208-850-1