International Journal on Advances in Internet Technology

2022 vol. 15 nr. 3&4

The *International Journal on Advances in Internet Technology* is published by IARIA.
ISSN: 1942-2652
journals site: http://www.iariajournals.org
contact: petre@iaria.org

Responsibility for the contents rests upon the authors and not upon IARIA, nor on IARIA volunteers, staff, or contractors.

Reference should mention:

> *International Journal on Advances in Internet Technology, issn 1942-2652*
> *vol. 15, no. 3 & 4, year 2022, http://www.iariajournals.org/internet_technology/*

Reference to an article in the journal is as follows:

> *<Author list>, "<Article title>"*
> *International Journal on Advances in Internet Technology, issn 1942-2652*
> *vol. 15, no. 3 & 4, year 2022, <start page>:<end page> , http://www.iariajournals.org/internet_technology/*

IARIA journals are made available for free, proving the appropriate references are made when their content is used.

Sponsored by IARIA
www.iaria.org

International Journal on Advances in Internet Technology
Volume 15, Number 3 & 4, 2022

Tzung-Shi Chen, National University of Tainan, Taiwan
Xi Chen, University of Washington, USA
IlKwon Cho, National Information Society Agency, South Korea
Andrzej Chydzinski, Silesian University of Technology, Poland
Noël Crespi, Telecom SudParis, France
Antonio Cuadra-Sanchez, Indra, Spain
Javier Cubo, University of Malaga, Spain
Sagarmay Deb, Central Queensland University, Australia
Javier Del Ser, Tecnalia Research & Innovation, Spain
Philipe Devienne, LIFL - Université Lille 1 - CNRS, France
Kamil Dimililer, Near East Universiy, Cyprus
Martin Dobler, Vorarlberg University of Applied Sciences, Austria
Jean-Michel Dricot, Université Libre de Bruxelles, Belgium
Matthias Ehmann, Universität Bayreuth, Germany
Tarek El-Bawab, Jackson State University, USA
Nashwa Mamdouh El-Bendary, Arab Academy for Science, Technology, and Maritime Transport, Egypt
Mohamed Dafir El Kettani, ENSIAS - Université Mohammed V-Souissi, Morocco
Armando Ferro, University of the Basque Country (UPV/EHU), Spain
Anders Fongen, Norwegian Defence Research Establishment, Norway
Giancarlo Fortino, University of Calabria, Italy
Kary Främling, Aalto University, Finland
Steffen Fries, Siemens AG, Corporate Technology - Munich, Germany
Ivan Ganchev, University of Limerick, Ireland / University of Plovdiv “Paisii Hilendarski”, Bulgaria
Shang Gao, Zhongnan University of Economics and Law, China
Emiliano Garcia-Palacios, ECIT Institute at Queens University Belfast - Belfast, UK
Kamini Garg, University of Applied Sciences Southern Switzerland, Lugano, Switzerland
Rosario Giuseppe Garroppo, Dipartimento Ingegneria dell'informazione - Università di Pisa, Italy
Thierry Gayraud, LAAS-CNRS / Université de Toulouse / Université Paul Sabatier, France
Christos K. Georgiadis, University of Macedonia, Greece
Katja Gilly, Universidad Miguel Hernandez, Spain
Mariusz Głąbowski, Poznan University of Technology, Poland
Feliz Gouveia, Universidade Fernando Pessoa - Porto, Portugal
Kannan Govindan, Crash Avoidance Metrics Partnership (CAMP), USA
Bill Grosky, University of Michigan-Dearborn, USA
Jason Gu, Singapore University of Technology and Design, Singapore
Christophe Guéret, Vrije Universiteit Amsterdam, Nederlands
Frederic Guidec, IRISA-UBS, Université de Bretagne-Sud, France
Bin Guo, Northwestern Polytechnical University, China
Gerhard Hancke, Royal Holloway / University of London, UK
Arthur Herzog, Technische Universität Darmstadt, Germany
Rattikorn Hewett, Whitacre College of Engineering, Texas Tech University, USA
Quang Hieu Vu, EBTIC, Khalifa University, Arab Emirates
Hiroaki Higaki, Tokyo Denki University, Japan
Dong Ho Cho, Korea Advanced Institute of Science and Technology (KAIST), Korea
Anna Hristoskova, Ghent University - IBBT, Belgium
Ching-Hsien (Robert) Hsu, Chung Hua University, Taiwan
Chi Hung, Tsinghua University, China
Edward Hung, Hong Kong Polytechnic University, Hong Kong
Raj Jain, Washington University in St. Louis , USA
Edward Jaser, Princess Sumaya University for Technology - Amman, Jordan
Terje Jensen, Telenor Group Industrial Development / Norwegian University of Science and Technology, Norway
Yasushi Kambayashi, Nippon Institute of Technology, Japan
Georgios Kambourakis, University of the Aegean, Greece

Jari Porras, Lappeenranta University of Technology, Finland
Neeli R. Prasad, Aalborg University, Denmark
Drogkaris Prokopios, University of the Aegean, Greece
Emanuel Puschita, Technical University of Cluj-Napoca, Romania
Lucia Rapanotti, The Open University, UK
Gianluca Reali, Università degli Studi di Perugia, Italy
Jelena Revzina, Transport and Telecommunication Institute, Latvia
Karim Mohammed Rezaul, Glyndwr University, UK
Leon Reznik, Rochester Institute of Technology, USA
Simon Pietro Romano, University of Napoli Federico II, Italy
Michele Ruta, Technical University of Bari, Italy
Jorge Sá Silva, University of Coimbra, Portugal
Sébastien Salva, University of Auvergne, France
Ahmad Tajuddin Samsudin, Telekom Malaysia Research & Development, Malaysia
Josemaria Malgosa Sanahuja, Polytechnic University of Cartagena, Spain
Luis Enrique Sánchez Crespo, Sicaman Nuevas Tecnologías / University of Castilla-La Mancha, Spain
Paul Sant, University of Bedfordshire, UK
Brahmananda Sapkota, University of Twente, The Netherlands
Alberto Schaeffer-Filho, Lancaster University, UK
Peter Schartner, Klagenfurt University, System Security Group, Austria
Rainer Schmidt, Aalen University, Germany
Thomas C. Schmidt, HAW Hamburg, Germany
Zary Segall, Chair Professor, Royal Institute of Technology, Sweden
Dimitrios Serpanos, University of Patras and ISI/RC ATHENA, Greece
Jawwad A. Shamsi, FAST-National University of Computer and Emerging Sciences, Karachi, Pakistan
Michael Sheng, The University of Adelaide, Australia
Kazuhiko Shibuya, The Institute of Statistical Mathematics, Japan
Roman Y. Shtykh, Rakuten, Inc., Japan
Patrick Siarry, Université Paris 12 (LiSSi), France
Jose-Luis Sierra-Rodriguez, Complutense University of Madrid, Spain
Simone Silvestri, Sapienza University of Rome, Italy
Vasco N. G. J. Soares, Instituto de Telecomunicações / University of Beira Interior / Polytechnic Institute of Castelo Branco, Portugal
Radosveta Sokullu, Ege University, Turkey
José Soler, Technical University of Denmark, Denmark
Victor J. Sosa-Sosa, CINVESTAV-Tamaulipas, Mexico
Dora Souliou, National Technical University of Athens, Greece
João Paulo Sousa, Instituto Politécnico de Bragança, Portugal
Kostas Stamos, Computer Technology Institute & Press "Diophantus" / Technological Educational Institute of Patras, Greece
Cristian Stanciu, University Politehnica of Bucharest, Romania
Vladimir Stantchev, SRH University Berlin, Germany
Tim Strayer, Raytheon BBN Technologies, USA
Masashi Sugano, School of Knowledge and Information Systems, Osaka Prefecture University, Japan
Tae-Eung Sung, Korea Institute of Science and Technology Information (KISTI), Korea
Sayed Gholam Hassan Tabatabaei, Isfahan University of Technology, Iran
Yutaka Takahashi, Kyoto University, Japan
Yoshiaki Taniguchi, Kindai University, Japan
Nazif Cihan Tas, Siemens Corporation, Corporate Research and Technology, USA
Alessandro Testa, University of Naples "Federico II" / Institute of High Performance Computing and Networking (ICAR) of National Research Council (CNR), Italy
Stephanie Teufel, University of Fribourg, Switzerland
Parimala Thulasiraman, University of Manitoba, Canada

Pierre Tiako, Langston University, USA
Orazio Tomarchio, Universita' di Catania, Italy
Dominique Vaufreydaz, INRIA and Pierre Mendès-France University, France
Krzysztof Walkowiak, Wroclaw University of Technology, Poland
MingXue Wang, Ericsson Ireland Research Lab, Ireland
Wenjing Wang, Blue Coat Systems, Inc., USA
Zhi-Hui Wang, School of Softeware, Dalian University of Technology, China
Matthias Wieland, Universität Stuttgart, Institute of Architecture of Application Systems (IAAS),Germany
Bernd E. Wolfinger, University of Hamburg, Germany
Chai Kiat Yeo, Nanyang Technological University, Singapore
Abdulrahman Yarali, Murray State University, USA
Mehmet Erkan Yüksel, Istanbul University, Turkey

International Journal on Advances in Internet Technology

Volume 15, Numbers 3 & 4, 2022

CONTENTS

Analyzing the Effects and Applicability of Social Media Elements in Notification Systems in Large Interconnected Organisations

Igor Jakovljevic
ISDS
Graz University of Technology
Graz, Austria
igor.jakovljevic@cern.ch

Christian Gütl
ISDS
Graz University of Technology
Graz, Austria
c.guetl@tugraz.at

Andreas Wagner
IT Department
CERN
Geneva, Switzerland
andreas.wagner@cern.ch

***Abstract*—Social media has become one of the most popular means of social interaction among humans, and recent statistics suggest that more than two thirds of Internet users use social media sites. This work investigates and determines aspects of social media that can be integrated into a notification system to minimize the effects of information overload. The preliminary findings of the study showed that introducing social media elements to notifications and notification systems potentially increases the credibility of notification systems and the clarity of notification. The evaluation consisted of the following parts: user demographics and general knowledge questionnaire, the execution of predefined tasks, rating of difficulty, and information provided by the system after the execution. It was carried out online with 35 students from Graz University of Technology and high school students from different schools in Austria and Kosovo. The participants reacted positively to notifications formatted as social media posts, rating them as more trustworthy than traditional notifications. Social media elements had the effect of helping the participants with determining the difference between fake and real information. The survey results could provide the initial steps toward new use cases of Social Media applications in notifications and other disciplines that deal with users' cognitive ability to process information or disciplines where information overload is considered detrimental. This research shows potential improvements to notification systems with the use of social media elements.**

***Index Terms*—Social Media; Notifications; Large Organisations; Hashtags; Microblogs**

I. Introduction

This paper is an extension of our previous work on the applicability of social media elements in notification systems in large interconnected organisations, presented in [1].

The use of modern information and communication technologies (ICT) has increased the amount of available information and made that information easily accessible. However, this has also resulted in users experiencing information overload [2], which can be defined as the state when users are presented with large amounts of information that exceed the processing capacity of the users [3].

One of the domains for information overload are notification systems. Users receive a large amount of notifications from multiple applications on multiple devices (i.e., an application that delivers to users information that they need to know through messages, e.g., a new email has been received). Notifications allow applications such as email clients, messaging applications, calendars, and others to inform users of incoming messages from other users, upcoming events, reminders, new emails, and more without explicitly requiring user interaction with the application. Since each application has a specific notification format, the user is presented with a large amount of different information, making it hard to process and keep an overview [4].

Based on a study of 40191 randomly selected participants from different areas of work, users receive on average 44.9 notifications per day from multiple sources. Participants received notifications from 173 applications. Some of the applications were email applications (e.g., Gmail or Outlook), text messaging applications (e.g., Whatsapp or SMS applications), and voice messaging applications (e.g., Google Hangout and Skype) [5]. Findings also shown that high number of notifications, in particular from email clients and social networking applications, correlate with increased stress and the feeling of being overwhelmed. They distract users from executing current tasks and induce negative emotions [6]. Another study based on a sample of high-performing management individuals has revealed that the increase of information overload leads to more stress and negative emotions in individuals [3]. Users recognize that notifications are potentially disruptive and distracting, as they disrupt the current engagement of the user [5].

Despite the disruptive nature of notifications, users decide to use them because of their benefit in providing relevant information. In this context, notification systems can be beneficial and attempt to aggregate the previously mentioned information from different sources (e-mail clients, news portals, messaging platforms, and others) and deliver it to the user in the form of notifications [7]. In addition to providing information aggregation and notification delivery, notification systems enable notification management (e.g., selecting which applications are allowed to send notifications), reducing the need of the user to constantly interact with different applications [8]. The success of a notification system depends on the accuracy of supporting the user with information between tasks, while simultaneously enabling utility by providing access to additional information [9]. Notification systems attempt to keep users informed by balancing the amount of valuable information provided and the disruption caused by the information. It is necessary to find means to coordinate the delivery of notifications from multiple applications across multiple devices or/and display only relevant information at a glance. By bringing together multiple sources of notifications, the user can determine the importance of a notification and reduce the level of distraction [10].

According to [9], there are three critical parameters for the creation of a successful notification system:

1) **Interruption** - is defined as an event where users have to shift their attention from the main task and switch focus to the notification. Examples of these events are receiving notifications while operating heavy machinery, where the notification should not distract the user from the main task. However, other situations, such as medical emergency alerts, require that the notification explicitly interrupts the user [11].
2) **Reaction** - is defined as the response to the stimulus provided by the notification. Some examples of user reactions are ignoring notifications, removing them from the notification list, and clicking on the notification.
3) **Comprehension** - is defined as the use of notification systems with the goal of remembering and making sense of information at a later point in time. Based on past reactions, a notification system can show notifications to users when they are more likely to read them.

While quick and correct reaction to information is important in many situations, it is also important to present the information in a comprehensible way. Notifications should display a balance between the interruption, reaction, and comprehension parameters based on situation, content and user habits and preference [9].

One of the main challenges with designing a notification system is learning when and how to display understandable and valuable messages at a glance without explicitly disturbing or distracting the user. This problem has been tackled in different disciplines. Potential practical concepts can be found on social media, especially Social Media Marketing (SMM). The goal in SMM is to present information to the user at a specific time based on previous user behavior and experiences with other similar users. The information contains social media elements and should not irritate the user but stimulate engagement with the content. This SMM information usually aims to guide the user to a social media site [12].

Social media sites have become one of the most popular means for social interaction among humans, and recent statistics suggest that more than two-thirds of internet users use social media sites [13]. One of the main reasons for its popularity is the user engagement and personalized information it provides to the users. There are also drawbacks such as the lack of security and privacy, internet addiction, frequent interruptions from other tasks, information overload, creation of information bubbles, and loss of social contacts [14].

Gamification is defined as the adoption of game technology and game design methods outside the games industry. Using game design elements in non-game contexts to motivate and increase user activity and retention has gained traction in diverse fields. Recent years have witnessed a rapid expansion of consumer software inspired by video games [15]. Motivated by gamification and its success we theorized that the reuse of elements of Social Media and SMM concepts in the field of notification systems could yield beneficial results. Concepts related to user engagement and information presentation can potentially be adopted in notification systems and other information systems to improve the flow of messages to users and improve user engagement.

Kietzmann et al. [16] identified seven main functional building blocks of social media: identity, conversations, sharing, presence, relationships, reputation, and groups. These building blocks can be identified in various social media applications, like networking sites, photo-sharing platforms, blogging platforms, video-sharing platforms, collaboration platforms, and micro-blogging platforms.

In this paper, we want to identify social media elements based on previously mentioned functional building blocks of social media and explore which of these social media elements can be adopted in a notification system. The goal is to improve user interaction and navigation, information value, information dissemination of notifications, and understanding of notifications in notification systems. Additionally, attention is also given to mitigate possible side effects of social media elements and notification systems, such as wasting time analyzing and reviewing information provided to the user through the notification system.

Based on the observations stated above, more specifically, the main research questions are:

- **RQ1**: Which elements of social media can be integrated into notification systems to display understandable and valuable notifications at a glance without explicitly disturbing the user?
- **RQ2**: Would users prefer to receive notifications with integrated social media elements like hashtags, topic keywords, source information, rating by other users, and groups information?
- **RQ3**: How do users react to notifications with additional information (hashtags, user group information, content approval/disapproval, and social media posts)?
- **RQ4**: Which emotions do users experience when receiving notifications with and without this additional information?

To this end, the remainder of this paper is organized as follows: Section II covers the literature overview and discusses current topics in social media, notification systems, and their relation and use-cases. In Section III, the methodologies used in the study and the study are explained. The results are presented in Section IV, together with a discussion of the study outcome. Finally, we conclude the work in Section V.

II. Background and Related Work

Inspired by SMM, where the integration of social media elements into marketing information has led to greater user engagement and satisfaction, we propose adopting social media elements in notification systems and notifications [12]. In analogy to gamification applying game design elements in non-game contexts [17], it is proposed to integrate social media elements in non-social media contexts. The application in notification systems aims to improve the readability of notifications and increase its information value.

The remainder of this section assesses the drawbacks and advantages of notification systems, describes social media elements, and investigates possible integration in notification systems.

A. Notification Systems

There are many different implementation versions of notification systems. The most commonly used are push notification systems for mobile phones, desktop status notification systems, browser-based notification systems, in-vehicle information systems, and others [18]. As mentioned in the previous chapter, notification systems attempt to communicate important information to users effectively without creating an unwanted intrusion into current user tasks [7]. Selecting important information for the user is a difficult task. A study of 400+ participants has shown that users are not satisfied with the notifications they receive from notification systems because they do not express the user's current interest. This leads to users ignoring most notifications from these systems [19]. Besides determining what is relevant information for the user, an essential concern in notification systems is the display of notifications without a significant interruption of users' main tasks. Visual implementations of notifications that typically are not a user's main attention priority are called secondary displays. Users willingly sacrifice brief interruptions from their primary task to view information of interest on these secondary displays [20].

There are several ways to display notification messages, and the state of the art can vary depending on the specific context in which the notifications are being used. Some common options for displaying notifications include using pop-up windows or banners on a computer or mobile device, using LED or visual indicators on hardware devices, or using in-app notifications within a mobile or web application. These technologies can be effective for alerting users to important information or events in a way that is timely and noticeable without being overly disruptive. Overall, the state of the art for displaying notification messages is constantly evolving, and there are many different technologies and approaches that can be used to effectively alert users to new information [19][20][21].

B. Social Media Elements

The above-mentioned functional building blocks of social media are umbrella terms used to cover many elements of social media observed on different social media platforms. Based on the analysis of social media sites and research on aspects of social media [12][16][22] we identified and summarized some of the most common elements. Table I displays these elements.

TABLE I
SUMMARY OF MAIN SOCIAL MEDIA ELEMENTS

Social Media Element	Description
Hashtags	A hashtag is a metadata tag type used on social networks to help users find resources with a specific theme or content [22][23]
Microblogs	Microblog services allow users to post and share short textual messages that are then propagated to an audience, which can then quickly interact with the posts and between each other [24]
Content approval/ disapproval	Social cues that send signals of social appropriateness or social acceptance of content to the content creator. Examples of these social elements are Likes, Retweets, Reactions, and more [25]
User Groups	User groups represent the extent to which users can form communities and sub-communities. The more 'social' a network becomes, the bigger the group of friends, followers, and contacts.
User-to-User Relationship	User-to-user relationships express the extent to which users can relate to each other (e.g., friendships on Facebook or Followers on Twitter) [16]
User Identity	It represents the degree to which users expose their identities on social media sites. It includes exposing information such as name, age, gender, profession, location, and other users' identifiable information [16].

Taking into account the definition of social media elements from Table I and the description of notification systems above, we have decided to exclude user identity from our research and for the review in this section. The main reason for the exclusion of this element is that it is too focused on the individual. Including user identity information in notifications displayed to the user does not improve the information on notifications. Showing this information would be redundant for the user and could not be integrated into the context of notifications without privacy concerns.

*1) **Microblogs**:* Similar to microblog posts, notifications are messages displayed to the users with the intent to share information. These messages can contain information from different applications (e.g., email subject and part of email text, new message alert). Based on the above description, it can be concluded that notifications share similarities to microblog post entries. However, unlike notifications that do not contain much additional information in their visual representation, microblog posts can contain aspects of social media, such as hashtags, group information, content approval/disapproval, and others. These elements allow the users to determine the importance and validity of a post. Aspects such as the number of individuals that have shared, liked, or approved the post, topics related to the shared post, and the type of individuals that have interacted with the post are of crucial importance to assess the value of the post and the information within [26] [27].

An example of a microblogging service is Twitter, one of the largest microblogging services with more than 300.000

posts generated daily. Twitter is classified as a social network because individuals can communicate and connect with each other to form social groups on Twitter. They form social groups by following each other or following trending hashtags and/or topics [24].

*2) **Hashtags**:* A hashtag is a metadata tag type that is used on social networks to help users find resources with a specific theme or content. The content of hashtags can be dynamically generated or user-generated and can only consist of letters, digits, and underscores. Hashtags are iconic features that enable easy retrieval of connected resources [22][23] . They are also used to construct a personal word/hashtag vector space of a user by examining the users' linguistic expression. Hashtags are inserted into the existing user word vector space using co-occurrence information and evaluated to determine whether the newly constructed vector space represents the personal linguistic expression of the individual [28]. These methods intend to represent individuals by learning about potential representations using hashtags. Different methods for learning semantic representations exist. Some of them are Word2Vec, Latent Semantic Analysis (LSA), Latent Dirichlet Allocation, and Recurrent Neural Network Language Model (RNNLM) [22]. Besides identifying and representing user characteristics, hashtags are used to connect similar resources, by assigning tags to provide contextual information [29].

Hashtag Retrieval is an information retrieval methodology which aims to retrieve relevant hashtags for a giver query from a collection of resources. Besides retrieval, an interesting topic for notifications is hashtag automatic annotation, where hashtags are generated based on content, with the goal to classify the content per topic. Automatic tag recommendation or annotation can improve the efficiency of text-based information retrieval systems. Due to the nature of hashtags it is possible to extract correlations between resources from different systems by exploring their hashtag representations [22].

*3) **Content approval/disapproval**:* There are several approaches to provide a system with the necessary user feedback information. IR systems use explicit information through user feedback or implicit information through user monitoring to determine user interests. Unobtrusive user interaction monitoring identifies content potentially interesting for users, without interfering with the user's normal work activity. Monitoring systems also leverage heuristics to deduce negative examples from observed behavior [30][31]. Providing and receiving feedback is also a fundamental component of participation in social media. In addition, the popularity of social media has enabled the use of rich user information from Facebook and other social networks to predict users' latent traits for recommendation [32]. Based on the study mentioned above, users have expressed a need for more personalization in notifications; integrating likes or dislikes into a notification system as a means of collecting feedback from the user related to notifications could be beneficial for improving the satisfaction rate of users [19].

*4) **User Groups**:* A widely discussed relationship group metric is the Dunbar Number, proposed by Robin Dunbar in 1992. He theorized that people have a cognitive limit that restricts the number of stable social relationships with other people to about 150. Social media platforms have recognized that many communities grow well beyond this number and offer tools that enable users management of memberships [33]. The assumption that the vocabulary used to discuss a topic stays similar between different user communities and does not vary significantly over time directly suggests that it is possible to compute the overlap of topics of two or more communities. This community similarity can connect communities from different social networks (e.g., Facebook), facilitate information sharing between communities, and extract community interest [34]. Furthermore, user groups and group behavior information infer social cues, including group information (e.g., number of people with the same interests who approved a notification or executed a specific action) in notifications could increase the credibility and information dissemination of notifications.

*5) **User-to-User Relationship**:* The type of relationships users form between each other determines what information exchanges between them. For example, when users form professional relationships online, the information exchanged between them will have professional content and high value, compared to friendly relationships where the information is of a different nature [16]. User relationship information could be used in notification systems to determine the character of information presented to the user.

C. Information Combination

As mentioned in [35], access to internal and external information and aggregation of different data sources is one of the main factors that increase the transparency, innovation, and productivity of large organisations. According to [36], linking information on Twitter with information from other sources like Wikipedia led to increased understanding of the information and productivity when consuming the information.

D. Privacy

The rapid growth of the Web has not only drastically changed the way people conduct activities and acquire information, but has also raised security and privacy issues for them. Users are increasingly sharing their personal information on social media platforms. These platforms publish and share user-generated data with third parties that risk exposing the privacy of individuals. Textual information is noisy, high dimensional, and unstructured. It is rich in content and could reveal many sensitive information that user does not originally expose such as demographic information and location [37].

E. Discussion

Towards our goal to determine how social media elements can enrich notifications with additional information, the section above outlined vital social media elements and investigated their application for this purpose. Table II summarizes

TABLE II
SOCIAL MEDIA ELEMENTS AND USABILITY IN NOTIFICATION SYSTEMS

Social Media Element	Usability in Notification Systems
Hashtags	Quick access to topic information; Enables instant classification of notifications by topic; Linking external information to the notification
Microblogs	Social Media Posts provide information representation ideas for notification due to their similarity; Content Sharing does not have a direct use in notifications
Content approval/disapproval	Provide a way for the user to express interest
User Groups	Provide additional information and credibility of information based on the opinion of a group of users
User-to-User Relationship	Provide different types of additional information based on relationships with different users

how social media elements could be beneficial for notification systems.

Hashtags and user group elements provide additional information, potentially enhancing the information in notifications. Integrating these elements could increase the trustworthiness of notification systems and reduce the time required for a user to evaluate the importance of notifications. Since notification systems lack a direct user feedback mechanism, integrating content approval/disapproval elements could provide it. Hashtags in Microblogging services contain information on the temporal trends of the information stream and the topology of the spread of information. This makes hashtags a suitable tool for archiving, tracking, and disseminating information [38][39]. Other applications of hashtags are advertising, indication of specific objects descriptions in posts or situations, expressing one's feelings. Interpreting the meaning of hashtags can be a means to learn potential semantic representations of words linked to hastags [28].

For microblogs, besides hashtags, our research focused on two additional features, Social Media Posts and Content Sharing. Due to the lack of applicability in notification systems, content sharing was excluded. However, considering that social media posts share similarities with notifications, we determined that formatting information in notifications similar to social media posts by including hashtags, more personalized text, and information sources could benefit notification systems.

Even though user-to-user relationships offer great insights into users' interests, knowing the user and connections are mandatory to integrate this element into a notification system. Due to the setting of our initial study, we excluded this element from the evaluation since it was necessary to track user relationships over a more extended period.

To this end, we have selected four social media elements for evaluation based on their applicability in notification systems: hashtags, user group information, content approval/disapproval, and social media posts (formatting the content of the notification as a social media post).

III. Research Study

The research study focused on providing insights and answering the previously mentioned research questions (RQ1-RQ4).

A. Study Design

To evaluate the influence of Social Media elements on notifications and notifications systems, it was necessary to conduct a user study that simulates user interaction with notifications and provides a way to display different types of notifications to participants. Measuring participant interaction with different notification types and enabling quantification of those interactions required the user study to pique participants' interest in cognitive tasks (reading and understanding articles) while showing notifications. The goal was to make the participants assume that the notifications were not part of the evaluation to receive non-biased results. Additionally, it was necessary to evaluate how well participants handled new concepts, which ones they preferred, and which ones they disliked. The evaluation consisted of the following parts: user demographics and general knowledge questionnaire, the execution of predefined tasks, rating of difficulty and information provided by the system after the execution of a task, questions for the System Usability Scale (SUS), questions for the Computer Emotion Scale (CES), Social Media Elements Importance Rating questionnaire, and a feedback questionnaire.

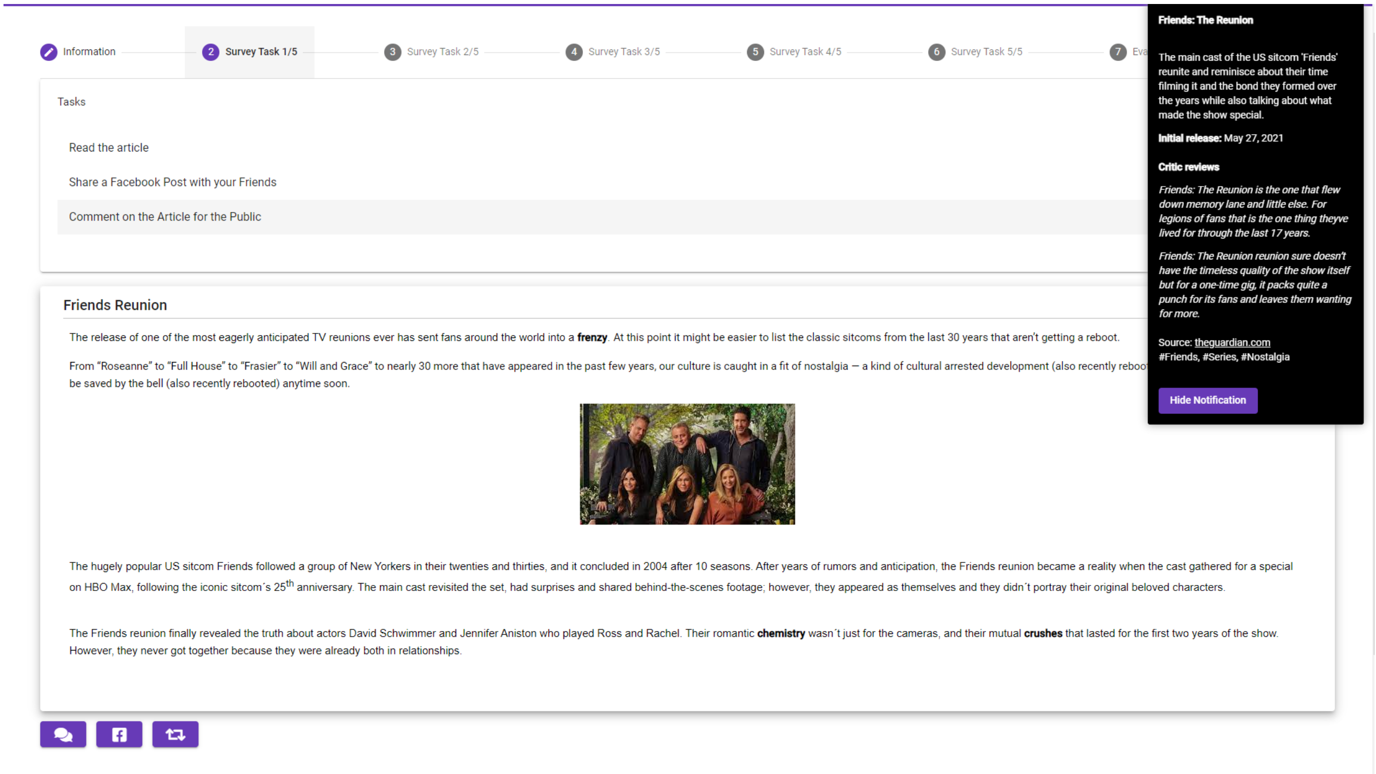
B. Settings and Instruments

The user study was executed online with students from the Graz University of Technology and high school students from different schools in Austria and Kosovo. It was designed as an AB study, which meant that the participants were separated into two groups (Group A and Group B). The participants were instructed to individually complete various tasks, after the execution of the tasks, they were asked to complete the previously mentioned questionnaires and specific survey questions.

*1) **General Questionnaire**:* contains questions listed in Table III that aim to identify the value and effects of additional information in notifications on the user. This questionnaire aims to provide insights to RQ1 and RQ2, by explicitly asking the participants about how they perceive SMEs in the notifications they received.

TABLE III
GENERAL QUESTIONNAIRE QUESTIONS

Question
Q1: Did you find the additional information in the notification valuable?
Q2: When I received notifications with additional information I was more confident in the notification?
Q3: Rank the additional information by importance
Q4: It was easier to understand the notification when I had additional information in the notification?
Q5: Did the notification break your concentration while executing the task?



Tasks

Read the article

Share a Facebook Post with your Friends

Comment on the Article for the Public

Friends Reunion

The release of one of the most eagerly anticipated TV reunions ever has sent fans around the world into a **frenzy**. At this point it might be easier to list the classic sitcoms from the last 30 years that aren't getting a reboot.

From "Roseanne" to "Full House" to "Frasier" to "Will and Grace" to nearly 30 more that have appeared in the past few years, our culture is caught in a fit of nostalgia — a kind of cultural arrested development (also recently reboo
be saved by the bell (also recently rebooted) anytime soon.

The hugely popular US sitcom Friends followed a group of New Yorkers in their twenties and thirties, and it concluded in 2004 after 10 seasons. After years of rumors and anticipation, the Friends reunion became a reality when the cast gathered for a special on HBO Max, following the iconic sitcom´s 25th anniversary. The main cast revisited the set, had surprises and shared behind-the-scenes footage; however, they appeared as themselves and they didn´t portray their original beloved characters.

The Friends reunion finally revealed the truth about actors David Schwimmer and Jennifer Aniston who played Ross and Rachel. Their romantic **chemistry** wasn´t just for the cameras, and their mutual **crushes** that lasted for the first two years of the show. However, they never got together because they were already both in relationships.

Friends: The Reunion

The main cast of the US sitcom 'Friends' reunite and reminisce about their time filming it and the bond they formed over the years while also talking about what made the show special.

Initial release: May 27, 2021

Critic reviews

Friends: The Reunion is the one that flew down memory lane and little else. For legions of fans that is the one thing theyve lived for through the last 17 years.

Friends: The Reunion reunion sure doesn't have the timeless quality of the show itself but for a one-time gig, it packs quite a punch for its fans and leaves them wanting for more.

Source: theguardian.com
#Friends, #Series, #Nostalgia

Fig. 1. Codis Survey Tool - Article Display with Notification

2) ***Article Feedback:*** contains questions listed in Table IV aimed to resolve if the users were able to determine and evaulate if the presented articles were fake or not. Since the survey was an AB survey, it is possible to use the feedback from this questionnaire to evaluate whether notifications with additional information help determine the truthfulness of articles and how users react to notifications with additional information.

TABLE IV
ARTICLE FEEDBACK QUESTIONS

Question
Q1: Do you think that the article ”Friends Reunion” is Fake or Real?
Q2: Do you think that the article ”Instagram for Children” is Fake or Real?
Q3: Do you think that the article ”People live in a 3D-Printed House” is Fake or Real?
Q4: Do you think that the article ”3 Reasons Why You Should Stop Eating Peanut Butter Cups!” is Fake or Real?
Q5: Do you think that the article ”Us Bacon Reserves Hit 50 Year Low” is Fake or Real?

3) ***Computer Emotion Scale (CES):*** The scale is used to assess the emotions of the participants, as it provides one of the most scientific ways to evaluate emotions. Anger, anxiety, happiness, and sadness are the emotions evaluated by the CES. The scale was used to answer RQ4 by determining the emotional influence of notifications on the participants, since it provides one of the most scientific ways for emotion evaluation [40].

4) ***System Usability Scale (SUS):*** The System Usability Scale (SUS) is used to measure the ease of use (EOU) of a system. It consists of ten items designed to assess EOU on a 100-point scale. Since its creation in the 1980s, SUS has been extensively used in human–computer interaction (HCI) research and practice to evaluate information technology (IT). It consists of a ten-item attitude likert scale that gives a global view of subjective assessments of usability scale. It is used to determine whether participants would prefer to receive additional information notifications, which is directly correlated with RQ2 and RQ3. It provides a trustworthy evaluation tool for usability testing [41].

5) ***SME Importance Rating:*** to determine which SMEs in Table II were important for understanding and perceiving notifications, participants were asked to rate the importance of the elements mentioned above, from 1 (not important at all) to 5 (Very Important),

6) ***Article Classification Questionnaire:*** was used to determine if users concluded that the read article was a real article or fake news. After reading all articles and receiving notifications related to the articles, participants were asked which articles they thought were fake news articles and which were real articles.

The study was created using the CoDiS Survey Tool [42]. The CoDiS Survey Tool is a web based evaluation tool which tracks and analyses participants' behavior while presenting specific assignments, displaying custom notifications, and displaying questionnaires to the participants. Figure 1 displays how the user interface of the CoDiS Survey Tool displays the articles, notifications and tasks for the participants. The user interface consists of a task view, where the participants can read the tasks related to the current article, seen in the top section of the image, above the title of the article and below the progress element. Below the article text are the user interaction buttons, that enable the user to execute article specific actions (e.g., share article on facebook, comment on article, etc.).

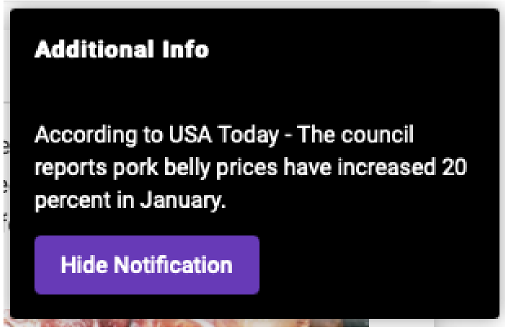
C. Procedure

Participants were asked to read articles mentioned in Table V and execute predefined tasks (share articles, comment on the article, and more).

TABLE V
ARTICLE TITLE AND VALIDITY

#	Title	Is Fake
1	Friends Reunion	No
2	People live in a 3D-Printed House	No
3	Instagram for Children	No
4	US Bacon Reserves Hit 50 Year Low	Yes
5	3 Reasons Why You Should Stop Eating Peanut Butter Cups!	Yes

As the participants completed these tasks, the notifications related to the articles were displayed. Notifications were displayed as part of the CoDiS Survey Tool as web elements that appear when the user starts reading an article. Depending on the user group, these notifications were either with additional information or without additional information. The additional information included hashtags, user group information, and social media post formatting. This additional information integrates all selected social media elements from the previous chapter.



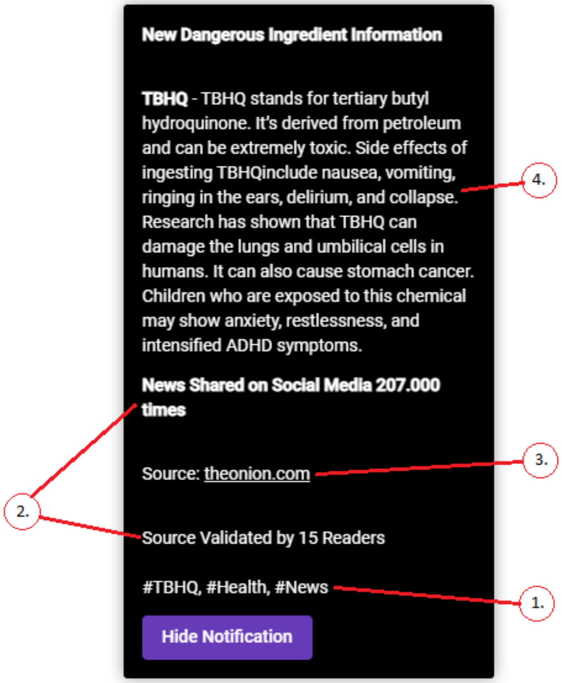
Additional Info

According to USA Today - The council reports pork belly prices have increased 20 percent in January.

Fig. 2. Simple Notification Information Display

Figure 2 displays a standard notification instance with only the notification text and action button. While Figure 3 previews a notification with additional information.

Hashtag display is marked with the number 1 on the figure, while the number 2 marks group information (e.g., number of readers that validated and/or shared the article). The notification text source is enumerated with number 3, and the notification text is marked with the number 4.



New Dangerous Ingredient Information

TBHQ - TBHQ stands for tertiary butyl hydroquinone. It's derived from petroleum and can be extremely toxic. Side effects of ingesting TBHQinclude nausea, vomiting, ringing in the ears, delirium, and collapse. Research has shown that TBHQ can damage the lungs and umbilical cells in humans. It can also cause stomach cancer. Children who are exposed to this chemical may show anxiety, restlessness, and intensified ADHD symptoms.

News Shared on Social Media 207.000 times

Source: theonion.com

Source Validated by 15 Readers

#TBHQ, #Health, #News

Fig. 3. Notification With Additional Information Display

D. Study Participants

The participant target groups for the study were high school and university students. In total, 215 individuals were asked to participate and only 35 completed the study. The age of the participants ranged from 15 to 34 years old, with 57.14% of the participants in the range from 15 to 20 years, 25.72 % in the range 20-25, 14.289 % in the range 25-30 and 2.85% in the range above 30 years old. Female participants made 28.57% of the total amount of participants, while male participants made 71.43%. As stated previously, the study was designed as an AB study, which is why the participants were divided into two groups (Group A and Group B). The purpose of this division is to reduce bias between users. Both groups received the first article with additional information notifications. The purpose of this was to create a control article and familiarize the users with this type of notifications. Group A received simple notifications on even-numbered articles, while group B received them on odd-numbered articles. After the participants finished reading the articles and the article-related tasks, they had to complete an evaluation.

IV. Findings and Discussion

Analyzing the answers to the questions presented in Table III, we have concluded that the participants find the notifications easier to understand and share the thought that they have more credibility when presented with additional information.

The additional information in notifications has increased the value of notifications to the user based on answers to Q1 from Table III. As seen in Table VI, 85.71% confirmed

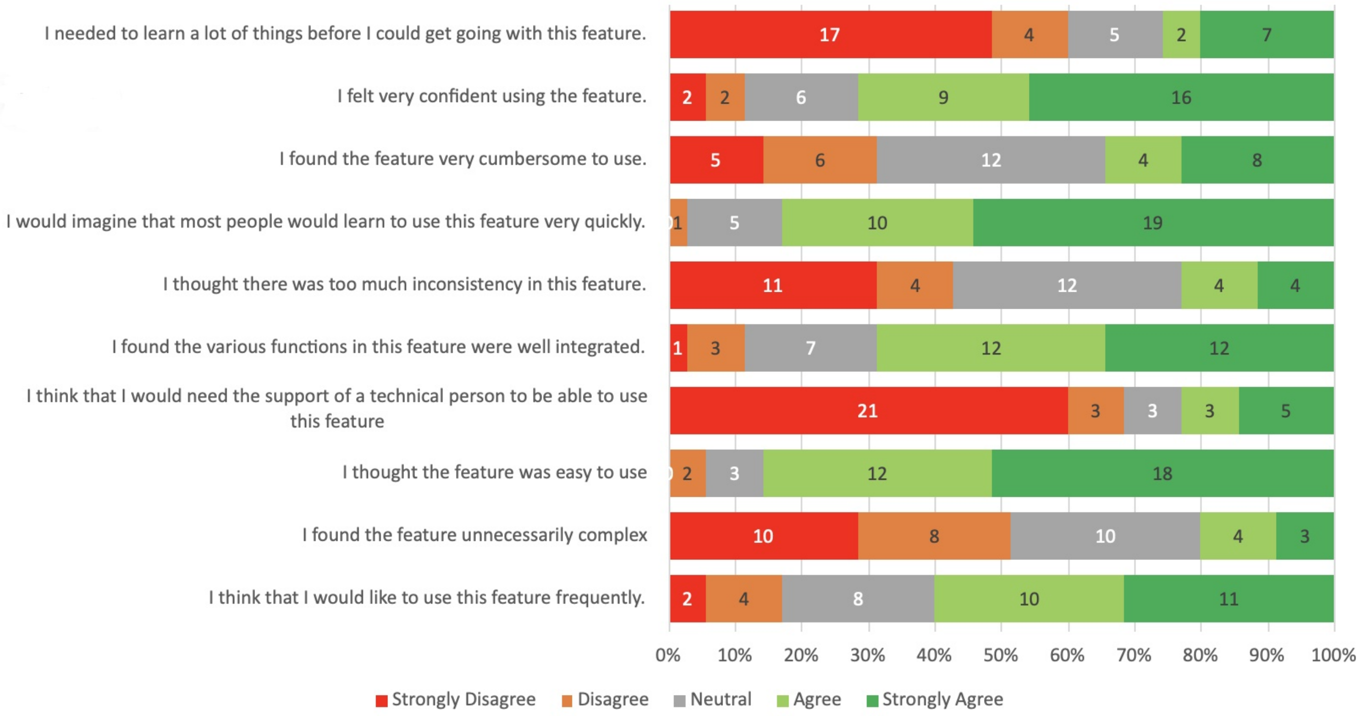


Fig. 4. System Usability Scale Detailed Results

TABLE VI
TABLE III QUESTIONNAIRE RESULTS

Question	Yes	No
Did you find the additional information in the notification valuable?	30 (85.71%)	5 (14.29%)
When I received notifications with additional information I was more confident in the notification?	21 (60.00%)	14 (40.00%)
It was easier to understand the notification when I had additional information in the notification?	27 (77.14%)	8 (22.86%)
Did the notification break your concentration while executing the task?	21 (60.00%)	14 (40.00%)

the premise that additional information is valuable to notifications. The participants had more confidence in notifications with additional information in comparison to standard notifications. Table VI shows that 60% of the participants voted that additional information increases the confidence of the notifications. Based on Q3 from Table III 77.14% of the participants stated that they find it easier to understand notifications with additional information. With 60% of the participants answering with ”Yes” to Q5 of Table III, we can confirm that the notifications break user concentration, which validates the results of previous research [3][5].

According to [9], the success of notification systems is dependent on the information they convey to the user. The survey participants agree with this as shown in Table VII. It reveals that users are predominantly concerned with the content and source of notifications. It implies that adding additional information to validate the content and source increases their value to users. The results in Table VII also validate our proposal that formatting notifications as social media posts could improve the information presented to the user since the content was formatted to be similar to a social media post. Contrary to our research, the group information (e.g., ”22 readers validated text”) was not ranked as highly important by the participants.

As seen in Figure 4, the distribution of SUS answers reveals that most of the users agree or strongly agree with questions Q2, Q4, Q6, Q8, and Q10 of the SUS [41] while disagreeing or strongly disagreeing with the rest of the questions. It is also visible that questions related to the negative rating of the system contain a significant portion of neutral answers, compared to questions focused on positive system ratings. These results indicate that the participants have formulated a positive opinion about notifications with additional information and that they would use a system with this feature, while not explicitly agreeing that there are negative aspects in such a system.

Due to the large number of neutral answers, the average rating of the SUS scale is 69.78. This is slightly above the limit of 68 set by [41] as the value that is the minimum for a usable system. Based on the SUS results, we can infer that users would prefer to use a social media notification system.

The result of the CES is shown in Table VIII, the table contains a list of feelings a participant has experienced. The CES shows that the users were happy most of the time executing tasks and receiving notifications, while none of the

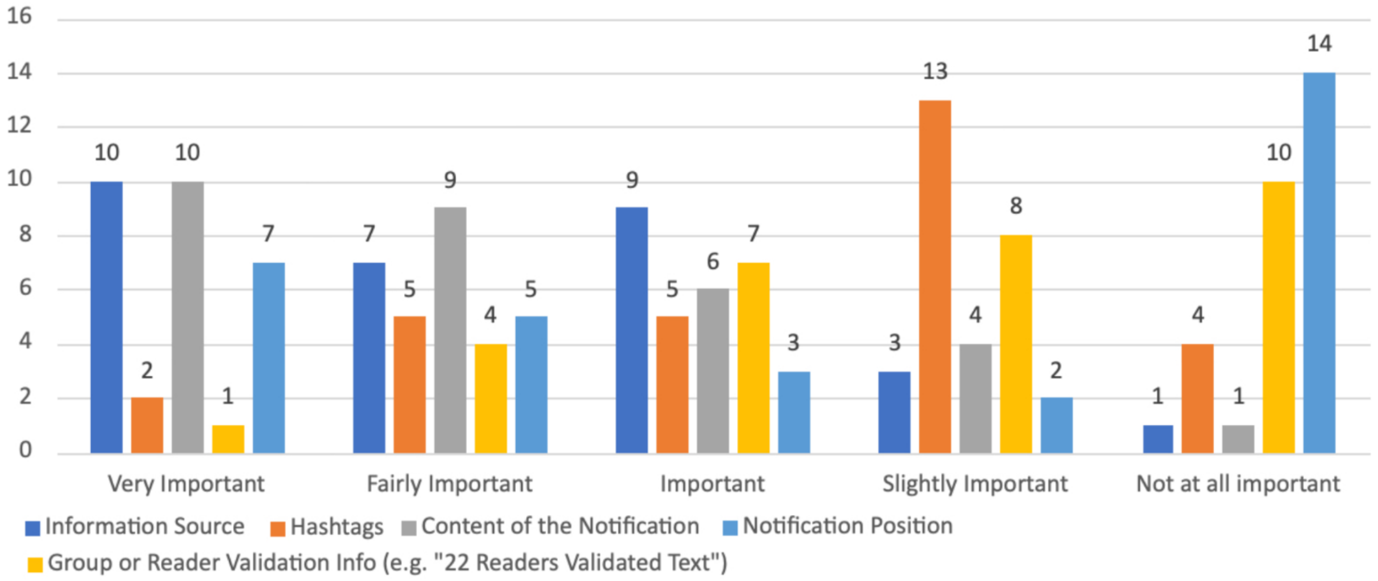


Fig. 5. Notification Element Ranking

TABLE VII
ADDITIONAL INFORMATION RANKING BY IMPORTANCE

Additional Information	Very Important	Not at all important
Information Source	10 (33.33%)	1 (3.33%)
Hashtags	2 (6.67%)	4 (13.33%)
Content of the Notification	10 (33.33%)	1 (3.33%)
Group or Reader Validation Info (e.g., ”22 Readers Validated Text”)	1 (3.33%)	10 (33.33%)
Notification Position	7 (23.33%)	14 (46.67%)

TABLE IX
ARTICLE VALIDITY EVALUATION RESULTS

Article Name	Fake News	Real Article
Friends Reunion	2 (6.90%)	27 (93.10%)
Instagram for Children	13 (44.83%)	16 (55.17%)
People live in a 3D-Printed House	15 (51.72%)	14 (48.28%)
3 Reasons Why You Should Stop Eating Peanut Butter Cups!	14 (48.28%)	15 (51.72%)
Us Bacon Reserves Hit 50 Year Low	13 (44.83%)	16 (55.17%)

time experiencing sadness, anxiety, and anger. According to Table VIII, the emotion anxiety has the lowest score because most users rated it with ”none of the time” followed by sadness and anger.

TABLE VIII
PERCENTAGE AN ANSWERS HAS BEEN SELECTED ON THE COMPUTER EMOTION SCALE

	None of the Time	Some of the Time	Most of the Time	All of the Time
Happiness	20.95%	31.43%	24.76%	22.86%
Sadness	68.57%	24.29%	4.29%	2.86%
Anxiety	69.29%	18.57%	8.57%	3.57%
Anger	65.71%	21.90%	7.62%	4.76%

The best-rated emotion was ”Happiness” where the majority of the users answered with either ”Some of the Time”, ”Most of the Time” or ”All of the Time”. These results do not correlate with previous studies, where users experienced negative emotions and stress while receiving notifications [6].

As part of the evaluations, the participants had to determine which articles were fake and which were real. The results of this evaluation are presented in Table IX. Besides the first article (”Friends reunion”), the participants could not distinguish fake from real. Only 57.93% of the cases were the articles correctly classified. Participants who received notifications with additional information classified articles with a 6.61% greater accuracy. As stated in Section III the participants were asked to rate the importance of the additional information, Figure 5 describes the result of the importance classification. The source of the information and the content of the notification were the elements that were rated as very important, while the position of the notification was classified as not at all important. Hashtags and group information were elements that were rated as important, leaning more towards slightly important than fairly important. In conclusion, the users appreciated information sources and content of notifications more than the position of notifications or user group validation information, while hashtags received a neutral rating.

Due to the inability to track the usage of notifications over a longer period, we could not evaluate all social media elements.

V. Conclusion

This work investigates and determines aspects of social media that can be integrated into a notification system to minimize the effects of information overload. With an emphasis on SME applications in notification systems, this research study concludes by demonstrating the potential of social media aspects in several fields. The preliminary analysis shows how the selected social media elements might enhance user satisfaction and the significance of information in notification systems. Additionally, it is observable that the additional SME information does not enhance the effects of information overload, but improves the perception and understanding of notifications. Based on the SUS it was determined that a system that uses SME for the enhancement of notifications is considered a s a usable system. Not all SMEs' use cases were investigated due to time restrictions and the limited number of SMEs considered. Additional SMEs, other media and devices for notification display could be considered for review in future work. Future research may also examine how users respond over time to messages that provide additional information. This could allow us to evaluate the SME studied and other SMEs that were unable to participate in this study more effectively. This may allow for a better evaluation of the analyzed SME and additional SMEs that could not be part of this study. Tracking user reactions for longer periods to different combinations of SMEs in notifications could lead to a novel approach to the use of SMEs within notification systems. The survey results could provide the initial steps toward new use cases of Social Media applications in notifications and other disciplines that deal with users' cognitive ability to process information or disciplines where information overload is considered detrimental. In conclusion, there are several advantages to integrating social media elements into notification systems. Using social media elements for notifications can help to ensure that important information is disseminated quickly and widely. This can be especially useful in emergencies, where time is of the essence. This research shows potential improvements to notification systems with the use of SME.

References

[1] I. Jakovljevic, A. Wagner, and C. Gütl, "Applicability of social media elements in notification systems in large interconnected organisations," in *The Eleventh International Conference on Social Media Technologies, Communication, and Informatics*, 2021, pp. 7–13, ISBN: 9781612088990.

[2] C. Gunaratne, N. Baral, W. Rand, I. Garibay, C. Jayalath, and C. Senevirathna, "The effects of information overload on online conversation dynamics," *Computational and Mathematical Organization Theory*, vol. 26, pp. 1–22, Jun. 2020. DOI: 10.1007/s10588-020-09314-9.

[3] P. G. Roetzel, "Information overload in the information age: a review of the literature from business administration, business psychology, and related disciplines with a bibliometric approach and framework developmen," *Business Research*, vol. 12, no. 2, pp. 479–522, Dec. 2019. DOI: 10.1007/s40685-018-0069-z.

[4] A. Visuri, N. van Berkel, T. Okoshi, J. Goncalves, and V. Kostakos, "Understanding smartphone notifications' user interactions and content importance," *International Journal of Human-Computer Studies*, vol. 128, pp. 72–85, 2019, ISSN: 1071-5819. DOI: https://doi.org/10.1016/j.ijhcs.2019.03.001.

[5] A. S. Shirazi, N. Henze, T. Dingler, M. Pielot, D. Weber, and A. Schmidt, "Large-scale assessment of mobile notifications," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '14, Toronto, Ontario, Canada: Association for Computing Machinery, 2014, pp. 3055–3064, ISBN: 9781450324731. DOI: 10.1145/2556288.2557189.

[6] S. T. Iqbal and E. Horvitz, "Notifications and awareness: A field study of alert usage and preferences," in *Proceedings of the 2010 ACM Conference on Computer Supported Cooperative Work*, ser. CSCW '10, Savannah, Georgia, USA: Association for Computing Machinery, 2010, pp. 27–30, ISBN: 9781605587950. DOI: 10.1145/1718918.1718926.

[7] D. S. McCrickard, M. Czerwinski, and L. Bartram, "Introduction: Design and evaluation of notification user interfaces," *International Journal of Human-Computer Studies*, pp. 509–514, 2003, ISSN: 1071-5819. DOI: https://doi.org/10.1016/S1071-5819(03)00025-9.

[8] M. Pielot, K. Church, and R. de Oliveira, "An in-situ study of mobile phone notifications," in *Proceedings of the 16th International Conference on Human-Computer Interaction with Mobile Devices Services*, ser. MobileHCI '14, Toronto, ON, Canada: Association for Computing Machinery, 2014, pp. 233–242, ISBN: 9781450330046. DOI: 10.1145/2628363.2628364.

[9] D. McCrickard, C. Chewar, J. Somervell, and A. Ndiwalana, "A model for notification systems evaluation—assessing user goals for multitasking activity," *ACM Trans. Comput.-Hum. Interact.*, vol. 10, pp. 312–338, Dec. 2003. DOI: 10.1145/966930.966933.

[10] D. Weber, A. S. Shirazi, and N. Henze, "Towards smart notifications using research in the large," in *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services Adjunct*, ser. MobileHCI '15, Copenhagen, Denmark: Association for Computing Machinery, 2015, pp. 1117–1122, ISBN: 9781450336536. DOI: 10.1145/2786567.2794334.

[11] D. C. Mcfarlane and J. L. Sibert, "Interruption of people in human-computer interaction," Ph.D. dissertation, 1998, ISBN: 0599230495.

[12] F. F. Li, J. Larimo, and L. Leonidou, "Social media marketing strategy: Definition, conceptualization, taxonomy, validation, and future agenda," *Journal of the Academy of Marketing Science*, vol. 49, pp. 51–70, Jun. 2020. DOI: 10.1007/s11747-020-00733-3.

[13] K. Kircaburun, S. Alhabash, Ş. Tosuntaş, and M. Griffiths, "Uses and gratifications of problematic social media use among university students: A simultaneous

examination of the big five of personality traits, social media platforms, and social media use motives," *International Journal of Mental Health and Addiction*, vol. 18, no. 3, pp. 525–547, 2020. DOI: 10.1007/s11469-018-9940-6. [Online]. Available: http://irep.ntu.ac.uk/id/eprint/33677/.

[14] M. Drahošová and P. Balco, "The analysis of advantages and disadvantages of use of social media in european union," *Procedia Computer Science*, vol. 109, pp. 1005–1009, 2017, 8th International Conference on Ambient Systems, Networks and Technologies, ANT-2017 and the 7th International Conference on Sustainable Energy Information Technology, SEIT 2017, 16-19 May 2017, Madeira, Portugal, ISSN: 1877-0509. DOI: https://doi.org/10.1016/j.procs.2017.05.446.

[15] S. Deterding, D. Dixon, R. Khaled, and L. Nacke, "From game design elements to gamefulness: Defining gamification," vol. 11, Sep. 2011, pp. 9–15. DOI: 10.1145/2181037.2181040.

[16] J. H. Kietzmann, K. Hermkens, I. P. McCarthy, and B. S. Silvestre, "Social media? get serious! understanding the functional building blocks of social media," *Business Horizons*, vol. 54, no. 3, pp. 241–251, 2011, ISSN: 0007-6813. DOI: https://doi.org/10.1016/j.bushor.2011.01.005.

[17] D. Dicheva, C. Dichev, G. Agre, and G. Angelova, "Gamification in education: A systematic mapping study," *Educational Technology Society*, vol. 18, pp. 75–88, Jul. 2015, ISSN: 1436-4522.

[18] D. McCrickard and C. Chewar, "Designing attention-centric notification systems: Five hci challenges," in *Cognitive Systems: Human Cognitive Models in Systems Design*, Psychology Press, 2006, ch. III, pp. 67–89.

[19] S. Pradhan, L. Qiu, A. Parate, and K. Kim, "Understanding and managing notifications," in *"IEEE INFOCOM 2017 - IEEE Conference on Computer Communications*, 2017, pp. 1–9. DOI: 10.1109/INFOCOM.2017.8057231.

[20] D. S. McCrickard, R. Catrambone, C. M. Chewar, and J. T. Stasko, "Establishing tradeoffs that leverage attention for utility: Empirically evaluating information display in notification systems," *International Journal of Human-Computer Studies*, vol. 58, no. 5, pp. 547–582, 2003, ISSN: 1071-5819. DOI: https://doi.org/10.1016/S1071-5819(03)00022-3. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1071581903000223.

[21] J. D. Lee, B. F. Gore, and J. L. Campbell, "Display alternatives for in-vehicle warning and sign information: Message style, location, and modality," *Transportation Human Factors*, vol. 1, no. 4, pp. 347–375, 1999. DOI: 10.1207/sthf0104_6.

[22] D. Correa and A. Sureka, "Mining tweets for tag recommendation on social media," in *Proceedings of the 3rd International Workshop on Search and Mining User-Generated Contents*, ser. SMUC '11, Glasgow, Scotland, UK: Association for Computing Machinery, 2011, pp. 69–76, ISBN: 9781450309493. DOI: 10.1145/2065023.2065040.

[23] J. Bieniasz and K. Szczypiorski, "Methods for information hiding in open social networks," *JUCS - Journal of Universal Computer Science*, vol. 25, no. 2, pp. 74–97, 2019, ISSN: 0948-695X. DOI: 10.3217/jucs-025-02-0074.

[24] H. Kwak, C. Lee, H. Park, and S. Moon, "What is twitter, a social network or a news media?" In *Proceedings of the 19th International Conference on World Wide Web*, ser. WWW '10, Raleigh, North Carolina, USA: Association for Computing Machinery, 2010, pp. 591–600, ISBN: 9781605587998. DOI: 10.1145/1772690.1772751.

[25] L. Scissors, M. Burke, and S. Wengrovitz, "What's in a like? attitudes and behaviors around receiving likes on facebook," in *Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work amp; Social Computing*, ser. CSCW '16, San Francisco, California, USA: Association for Computing Machinery, 2016, pp. 1501–1510, ISBN: 9781450335928. DOI: 10.1145/2818048.2820066.

[26] M. Efron, "Hashtag retrieval in a microblogging environment," in *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '10, Geneva, Switzerland: Association for Computing Machinery, 2010, pp. 787–788, ISBN: 9781450301534. DOI: 10.1145/1835449.1835616.

[27] F. Zamberi, N. Adli, N. Hussin, and M. Ahmad, "Information retrieval via social media," *International Journal of Academic Research in Business and Social Sciences*, vol. 8, pp. 1375–1381, Jan. 2018. DOI: 10.6007/IJARBSS/v8-i12/5239.

[28] S. Seo, J. Kim, S. Kim, J. Kim, and J. Kim, "Semantic hashtag relation classification using co-occurrence word information," *Wirel. Pers. Commun.*, vol. 107, no. 3, pp. 1355–1365, 2019. DOI: 10.1007/s11277-018-5745-y.

[29] A. Belhadi, Y. Djenouri, J. C. Lin, C. Zhang, and A. Cano, "Exploring pattern mining algorithms for hashtag retrieval problem," *IEEE Access*, vol. 8, pp. 10 569–10 583, 2020. DOI: 10.1109/ACCESS.2020.2964682.

[30] S. Middleton, N. Shadbolt, and D. De Roure, "Ontological user profiling in recommender systems," *ACM Trans. Inf. Syst.*, vol. 22, pp. 54–88, Jan. 2004. DOI: 10.1145/963770.963773.

[31] J. Das, P. Mukherjee, S. Majumder, and P. Gupta, "Clustering-based recommender system using principles of voting theory," in *2014 International Conference on Contemporary Computing and Informatics (IC3I)*, 2014, pp. 230–235. DOI: 10.1109/IC3I.2014.7019655.

[32] S. Sedhain, S. Sanner, D. Braziunas, L. Xie, and J. Christensen, "Social collaborative filtering for cold-start recommendations," in *Proceedings of the 8th ACM Con-*

ference on Recommender Systems, ser. RecSys '14, Foster City, Silicon Valley, California, USA: Association for Computing Machinery, 2014, pp. 345–348, ISBN: 9781450326681. DOI: 10.1145/2645710.2645772. [Online]. Available: https://doi.org/10.1145/2645710.2645772.

[33] W.-X. Zhou, D. Sornette, R. A. Hill, and R. I. M. Dunbar, "Discrete hierarchical organization of social group sizes," *Proceedings of the Royal Society B: Biological Sciences*, vol. 272, no. 1561, pp. 439–444, Feb. 2005, ISSN: 1471-2954. DOI: 10.1098/rspb.2004.2970.

[34] P. Lorenz-Spreen, F. Wolf, J. Braun, G. Ghoshal, N. Djurdjevac-Conrad, and P. Hövel, "Tracking online topics over time: Understanding dynamic hashtag communities," *Computational Social Networks*, vol. 5, pp. 5–9, 2018. DOI: 10.1186/s40649-018-0058-6.

[35] C. G. Igor Jakovljevic and A. Wagner, "Open search use cases for improving information discovery and information retrieval in large and highly connected organizations," *2nd Open Search Symposium*, 2020. DOI: 10.5281/zenodo.4592449.

[36] P. Dooley and B. Božić, "Towards linked data for wikidata revisions and twitter trending hashtags," in *Towards Linked Data for Wikidata Revisions and Twitter Trending Hashtags*, New York, NY, USA: Association for Computing Machinery, 2019, pp. 166–175, ISBN: 9781450371797. DOI: 10.1145/3366030.3366048.

[37] G. Beigi and H. Liu, "A survey on privacy in social media: Identification, mitigation, and applications," *ACM/IMS Trans. Data Sci.*, vol. 1, no. 1, Mar. 2020, ISSN: 2691-1922. DOI: 10.1145/3343038. [Online]. Available: https://doi.org/10.1145/3343038.

[38] O. Tsur and A. Rappoport, "What's in a hashtag? content based prediction of the spread of ideas in microblogging communities," in *WSDM 2012 - Proceedings of the 5th ACM International Conference on Web Search and Data Mining*, May 2012, pp. 643–652. DOI: 10.1145/2124295.2124320.

[39] S. Fedushko, Y. Syerov, and S. Kolos, "Hashtag as way of archiving and distributing information on the internet," in *Modern Machine Learning Technologies, Workshop Proceedings of the 8th International Conference on "Mathematics. Information Technologies. Education"*, ser. CEUR Workshop Proceedings, vol. 2386, 2019, pp. 274–286. [Online]. Available: https://ceur-ws.org/Vol-2386/paper20.pdf (visited on 05/11/2022).

[40] R. Kay and S. Loverock, "Assessing emotions related to learning new software: The computer emotion scale," *Computers in Human Behavior*, vol. 24, pp. 1605–1623, Jul. 2008. DOI: 10.1016/j.chb.2007.06.002.

[41] A. Bangor, P. Kortum, and J. Miller, "Determining what individual sus scores mean: Adding an adjective rating scale," 3, vol. 4, Bloomingdale, IL: Usability Professionals' Association, May 2009, pp. 114–123.

[42] I. Jakovljevic, "Codis survey tool," DOI: 10.5281/zenodo.5345121. [Online]. Available: https://zenodo.org/record/5345121 (visited on 05/11/2022).

A Peer-Teaching Support System for Online Exercises: Prototype and its Evolution

Shiryu SEKIGUCHI
Electrical Engineering and Computer Science Course
Shibaura Institute of Technology
Tokyo, Japan
ma21078@shibaura-it.ac.jp

Tsuyoshi NAKAJIMA
Electrical Engineering and Computer Science Course
Shibaura Institute of Technology
Tokyo, Japan
tsnaka@shibaura-it.ac.jp

***Abstract*—Online exercises that deal with hardware, such as Internet of Things prototyping, have few convenient tools for students to share visual information on physical artifacts for peer-teaching. We proposed a peer-teaching support system "ShareHandy," which allows students to share videos of artifacts under development and point out areas of interest in real time using smartphones and PCs. We developed a prototype and applied it to an online Internet of Things prototyping exercise and verified its effectiveness for peer-teaching. As a result, the students could understand each other's problems and give appropriate advice efficiently and accurately. This proves that the system is effective for peer-teaching and improving the overall efficiency of the exercise. However, we found that it has little effect on promoting peer-teaching itself. Based on the findings, we defined the requirements for improved system necessary to promote peer-teaching in the exercises. This paper describes and evaluates the first protype of ShareHandy and provides the requirements for its evolution.**

Keywords-online education; peer-teaching; IoT exercise.

I. Introduction

Peer-teaching is a learning method for students to teach each other, gaining attention as a way to increase the effectiveness of student learning [2]. Recently, the COVID-19 pandemic has increased the number of online classes and exercises. However, such an online situation makes peer-teaching difficult because of the limited interactions between students [4]. One of the main reasons for this is that it is difficult for students to share the information on their physical artifacts under development visually. For example, in an online exercise where students learn to assemble the same electric circuit, they lose important peer-teaching opportunities, such as sharing one student's work with the teacher and other students, pointing out and giving advice to each other interactively, and reviewing the advised points after that.

To solve this problem, Dorneich et al. [5] provide several best practices to promote effective online team collaboration. Kitagami et al. [6] propose an exercise method of using a virtual camera to share camera images as well as documents, on one screen in a Web conference system, in which students watch the screen image transmitted by the teacher as they proceed with the exercises. However, this method does not allow students to see other students' artifacts while receiving teachers' transmission. Hamblen et al. [7] proposed a method for peer-teaching of hardware assembly in an asynchronous environment using a wiki. Students can interact with each other using the wiki, but only in a non-real-time manner.

We propose a peer-teaching support system "ShareHandy" to solve the problems mentioned above to ease online peer-teaching in a group of students. In the previous work [1], we developed a prototype of the system, which has several functions to share students' physical artifacts visually with each student's smartphone or Web camera: pointing at and drawing on the shared videos, saving the snapshot of the videos with the drawings.

This paper fleshes out the contents of the previous paper to provide detailed discussion, presenting the requirements for the peer-teaching support system, its design and implementation corresponding to the requirements, the evaluation of its usefulness by applying it to a group exercise of Internet of Things (IoT) prototyping held online, and the proposal of additional functions to promote peer-teaching like progress sharing and document pointer.

In this paper, Section 2 presents the details of our proposal system, Section 3 describes the experimental evaluation, results and discussion, Section 4 presents the requirements for its evolution, and Section 5 concludes this paper.

II. Proposed System

To support peer-teaching in IoT prototyping exercises, it is necessary to have a support system for the students (and the teacher) to share the students' physical artifacts visually. The proposed system provides stream live videos of students' artifacts using their own smartphone or Web cameras, which they can point at and draw additional information on.

The following subsections describe usage scenarios of the proposal system, its functions, and how the functions can be utilized in the scenarios.

A. Usage Scenarios

The target exercise assumes a situation in which each student work individually on the same assignment and several small groups of students are organized for peer-teaching. The students can use the proposed system to discuss and teach each other in the group. In this situation, students may frequently switch their role between tutors (teaching) and learners (being taught).

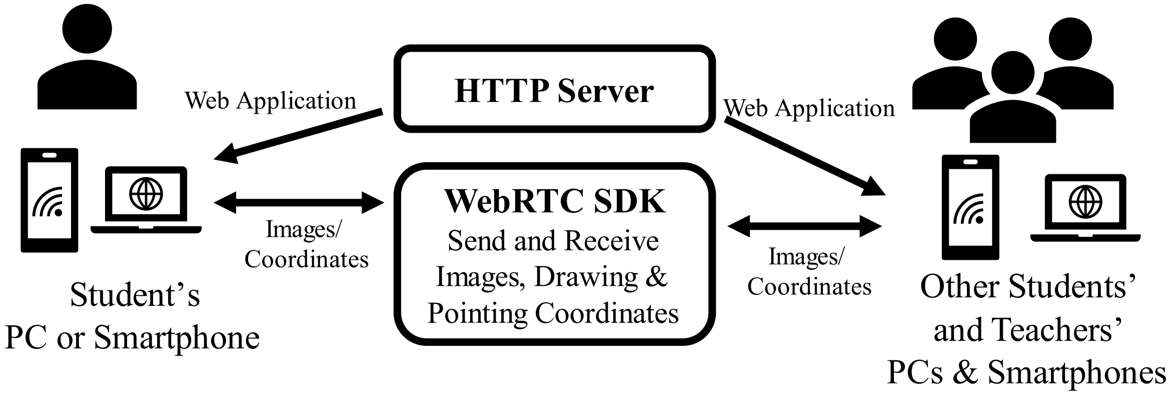


Figure 1. System Structure of ShareHandy

The proposed system is supposed to be used with the Web conferencing system, which handles document sharing and voice communication. The system helps the group members share video of their artifacts under development.

We assume the following usage scenarios of the system:

- Students who have successfully completed the assignment and those who have not finished yet, compare each other's artifacts to find out what to do for solving the problems the no-completed students have.
- A student asks the teacher of his questions, and a session is held to answer them. Some other students interested in the questions participate in the session, and the session proceeds on a question-and-answer basis.

B. *Designed functions and how to use in the scenarios*

We listed the designed functions and how it will work for above scenarios below.

1) Setting up rooms for group sharing

The system allows users to set up multiple rooms at a time so that users can easily have a space to discuss some topic, which other users can participate in freely when they are interested in the topic.

2) Real-time video streaming

Images of multiple students' work-in-progress artifacts are streamed in real-time by using smartphones or Web cameras attached to laptops. The system can display multiple videos simultaneously so that the students can see and compare each other's artifacts. This allows the tutors to convey, what changes are supposed to occur in each of the correct steps and allows the learners to explain what situation they are in and what their problems are. In addition, the system helps the users recognize the artifacts in 3D, which is not possible with documents or still images.

3) Pausing video and broadcasting still images

The video streaming can be paused to share as a still image. This allows the tutors to have enough time for detailed explanation at the key points and allows the learners to stop at their problematic points to ask for advises.

4) Pointing by participants

Each user has a cursor with a personal identifier, which can be placed at a specific position on the video image to indicate his or her point of interest. This allows the teaching student to clarify the focus points, explain the problems, and give instructions accurately. It also allows the person being taught to show the points to ask.

5) Writing and drawing by users

Users can write texts and draw figures on the streaming videos (including still images) using pointing devices. This allows the tutors to label related locations (e.g., initials and numbers) or to indicate directions by drawing arrows. In addition, the learners can leave notes to be shared.

6) Saving video captures and drawings as still images

Still images of videos with drawing can be stored on the user's device. This allows the learners to review the session afterward.

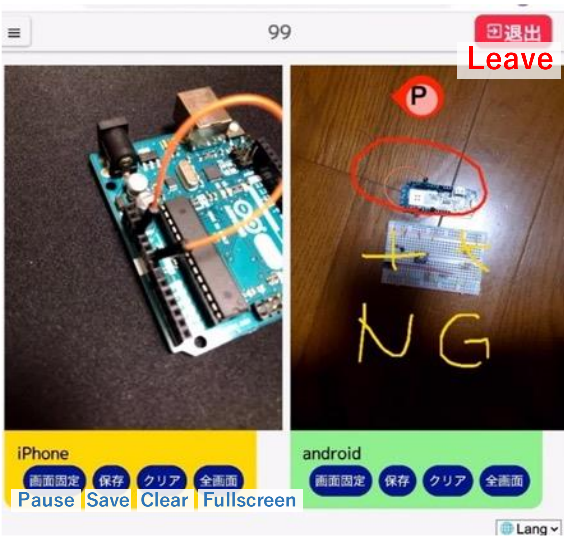


Figure 2. User Interface of ShareHandy and Demonstration

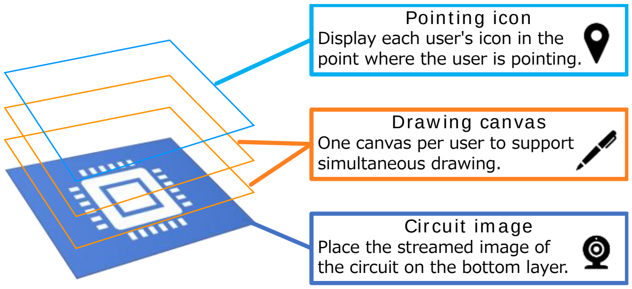

Pointing icon
Display each user's icon in the point where the user is pointing.
Drawing canvas
One canvas per user to support simultaneous drawing.
Circuit image
Place the streamed image of the circuit on the bottom layer.

Figure 3. Layered Canvas Function

C. *Implementation of the Proposal System*

The proposed system has been implemented as a prototype first. The system structure of the prototype is shown in Figure 1.

The prototype is built as a Web application [8], which runs on a Web browser and has the advantage to support multiple operating systems with a single code. This allows it to work on devices like PCs and smartphones, in which users only are required to access a web page to use the application. Moreover, there is no need to install additional applications on the device.

As shown in Figure 1, we use WebRTC (Web Real-Time Communication), which is a platform to enable real-time communication of video, audio, and general data via a Web browser [9]. WebRTC communication allows the users to send and receive videos and images, coordinates of drawings and pointers, and other control signals in real-time.

Figure 2 shows an example of using the prototype in real. Three users participate in the session, and two members share videos showing their artifacts. On the left screen, the pointers of the users (shown with a “P” mark) and drawing data are displayed (shown with the text of “NG”). In this situation, the student shown green needs help, and yellow and red members help him. The drawing color and pointing color are the same as the nametag shown at the bottom of the videos, and students choose their color at the beginning of a session.

Table I. QUESTIONS ON THE QUESTIONNAIRE ADMINISTERED AT THE END OF THE EXPERIMENT

No.	Questions and Options
Q1	What is your position?
Q2	In what situations did you use the system? ・ Teaching each other in the group ・ Creating something with the group member ・ Teaching each other with one-on-one
Q3	Please answer the following questions in four choices: Agree, Almost Agree, Almost Disagree, or Disagree
Q3-1	It was easy to see the other person's hand.
Q3-2	Groupmates understood my situation right away.
Q3-3	I was able to work efficiently.
Q3-4	It was easy to communicate by pointing at the hand.
Q3-5	It lowered the barrier for teaching each other.
Q3-6	I would use this system again.
Q4	Any other comment (Free text)

To draw and point with some members simultaneously in real-time is difficult with one canvas, so we provide the drawable sections by overlaying multiple canvases on a video, which are the same number as the group members. This implemented as Layered Canvas Function, which keeps drawings and images independently and overlay them to display and save as an image, as shown in Figure 3.

III. EXPERIMENTAL EVALUATION

To evaluate the proposed system in Section 2, we conducted experiments by using online exercises.

A. *Purpose of the Experiment*

The following two research questions are set concerning the effects when students use the prototype in online exercises.

[RQ1] Does this system allow the tutors to convey correct steps quickly and accurately to be done on the artifact to the students taught?

[RQ2] Does this system enable group communication to be carried out smoothly, and as a result, it can promote peer-teaching?

We conducted experiments to investigate these research questions.

B. *Method of the experiments*

We conducted the experiments with following method.

1) *Exercises to be applied*

The prototype system is applied to the following two real online exercises, individual exercise and group project-based exercise. Individual exercise is the one following a sample procedure, where every student belongs to some group whose members will communicate within it. Group exercise is Project-based learning style one, in which each group develops a prototype to solve their problem.

2) *Subjects' attributes and experimental procedure*

We held the experiments with ten undergraduate students and four graduate school students. All subjects were students at Shibaura Institute of Technology and majored in computer science and engineering. We formed the subjects to small groups of three to four people, and we asked the groups to use the prototype freely during the exercises. We did not instruct where and how to use it.

3) *Evaluation method*

The subjects are asked to answer to several questions on a questionnaire after finishing the exercises. Their answers are analyzed to evaluate the results of the experiments. The questions in the questionnaire are shown in Table 2. In addition to this, the subjects are directly interviewed after the exercises.

C. *Result of the Experiment*

We received 14 answers from all the subjects. The results of the questionnaire are shown in Figure 4.

- Concerning the answers to Q3-1, 86% of the subjects answered positively to the question, “Was it easy to see the other person’s hand?” In addition, all the subjects felt positive about Q3-2, “The other person quickly understood my situation,” and Q3-4, “It was

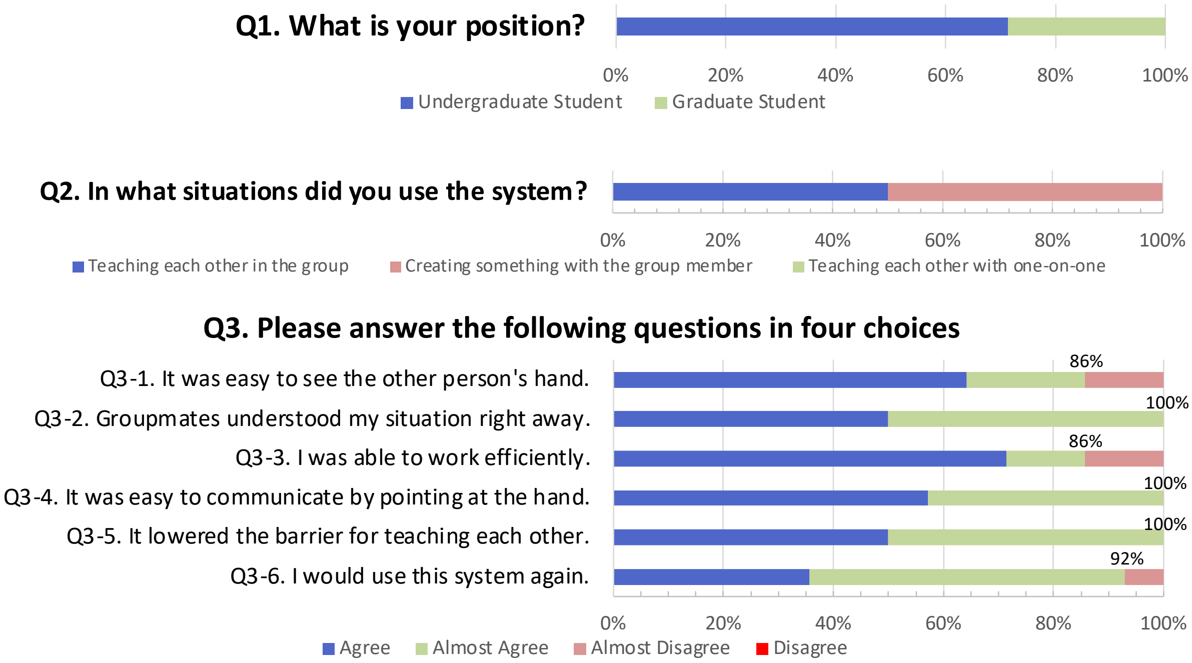


Figure 4. Questionnaire Items & Result of Answers

easy to communicate by pointing with my hand." Furthermore, 86% of the subjects answered positively to Q3-3, "I was able to work efficiently." From these results, we can confirm that the system works as expected and allows the subjects to convey instructions on artifacts efficiently and accurately.

- To Q3-5 of "It lowered the barrier for teaching each other," all the subjects answered affirmatively.

In addition, 92% of the students answered the question affirmatively if they would like to use this system again, suggesting that using this system is helpful.

From the interviews with the students after the experiment, we obtain the following points of their impressions and dissatisfaction.

Positive comments include:

- It is easy to understand how the artifacts are built because they can be easily seen, and their assembling steps can be understood well. (Answered from many subjects).
- Because group members can share their artifacts visually, it is easy to explain them. Some members used it to check if the correct sensor has been selected.

The second comment suggests that we can apply the proposed system to much broader things than we initially expected.

However, we have some negative comments:

- The screen of the smartphone is too small to see the screens of other participants.
- The camera does not focus well, and it is difficult to see small letters.

These comments suggest a need for improvement in both the prototype itself and how to use it (Issue 1).

D. Discussion of the result of the experiments

We found that the proposed system effectively achieves the effects that we targeted in the design of the system from the answers of Q3-1 to Q3-4. Therefore, the system allows all the subjects (including teachers and students) to share multiple hardware visually in real-time and help them communicate (RQ1). Moreover, based on the answers to Q3-5, it can be directly found that the system lowers the barrier for peer-teaching (RQ2).

The overall results from the answers to the questionnaire show the level of satisfaction in using this system is high. This is mainly because there is no other way for students to see others' artifacts and pointing at them in real-time other than the system.

However, the system has not been used often during the exercises, i.e., the frequency to use was lower than expected. Some student commented about it:

- The system is easy to use, but we did not have so many opportunities to use it.
- We did not have so many problems relating to hardware in IoT prototyping.

The main reason for it is that the IoT prototyping used in the experiments does not have so many complicated tasks relating to hardware to require someone's help.

In addition, we found from some interview results that online exercises tend to create an emotional distance between

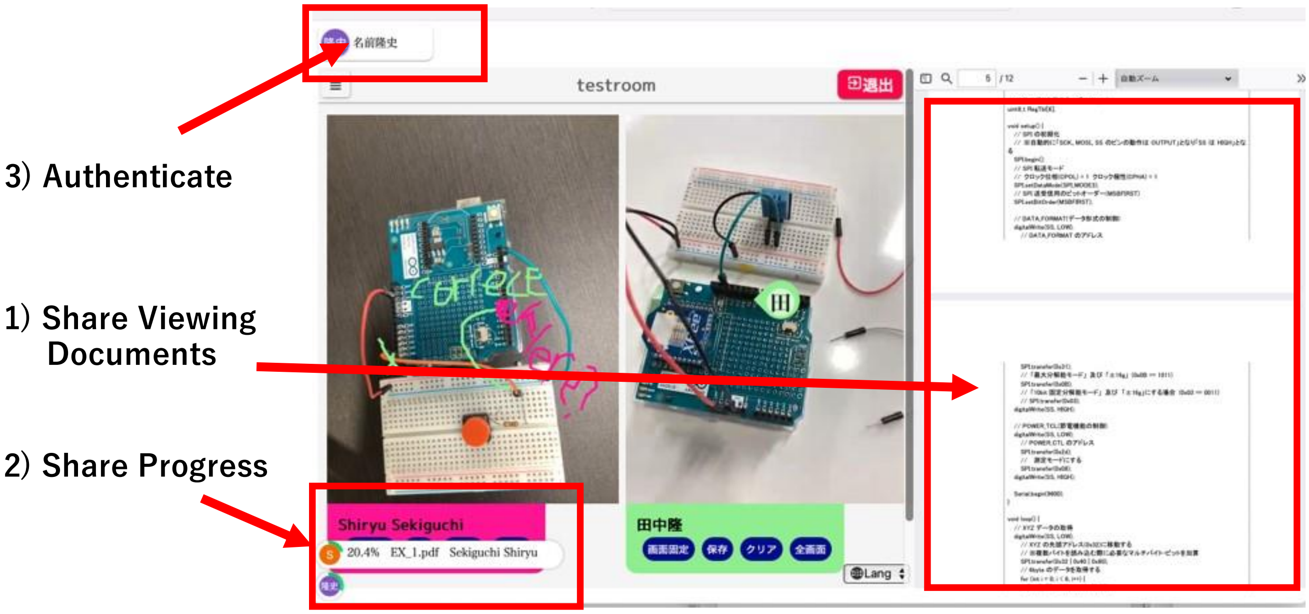


Figure 5. Image of the prototype of Integrated ShareHandy

students because it does not provide the opportunities to know what the other students have done and are doing now (Issue 2). For this reason, many students are hesitant to ask questions of the other group members and instead possibly chose to solve the problems for themselves.

We concluded through the experiment that further system improvements are needed to reduce students' hesitancy to communicate online in order to close the distance between minds.

IV. Improvements of the Proposed System

Through the experiments of the system, we obtained the results and identified the problems of the proposed system from the questionnaire and the interview. Among them, we focused on the following strategies to activate peer-teaching from the issues showed Section 4.

S1. Reduce emotional distance between members for peer-teaching.

S2. Improve usability [10] when using the system.

A. Approach to improving the proposed system

To settle the issues mentioned above, we are planning to develop "Integrated ShareHandy" that comprehensively supports peer-teaching in online exercises. This improvement aims to increase the frequency of use of the system by groups of students by reducing emotional barriers to communication.

For this goal, we set the following additional use case:

- Students who have completed the exercise and students who are stuck on some problems use the system. They go through the exercise to follow the tutors' steps to check each other's work done step by step.

B. New functions to reduce the emotional distance

This subsection describes three functions to be required for Integrated ShareHandy.

1) Sharing exercise documents

Function: All users share referenced documents, which they can compare it with the video of their artifacts, drawing on and pointing at the documents.

This function is expected to enhance the effectiveness of peer-teaching by allowing all the users being taught simultaneously to see the same document to compare it with their artifacts to find out the problems. This also aims to improve the understanding of those being taught.

2) Sharing the progress

Function: The system collects and displays students' progresses.

This function helps students notice some others' delay in progress in the group early, providing them for the opportunities for peer-teaching. The students to be supported can recognize who can help, asking them for support. They can also recognize who has the same level of progress, calling for teaching each other to solve the same problems. In addition, students to provide support can recognize students who probably need help, actively giving offer support to them (S1 in Section 4). We believe that this will increase the frequency to use the system for peer-teaching.

As the other effect of this function, it enables teachers to check students' progress to provide appropriate support.

3) Authentication for students

Function: Users can link their multiple logins under their own account IDs to participates in a meeting.

This is convenient when one user logins with multiple devices to participate in a meeting, such as a smartphone for

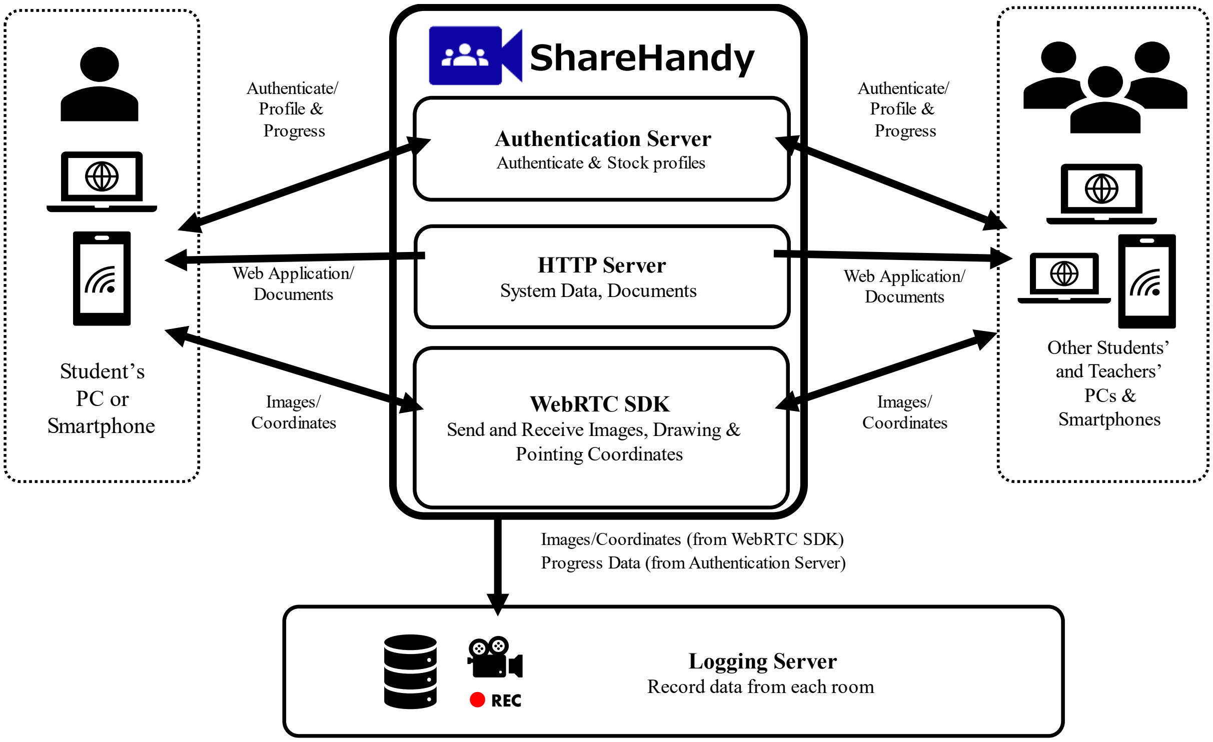

ShareHandy
Authentication Server
Authenticate & Stock profiles
HTTP Server
System Data, Documents
WebRTC SDK
Send and Receive Images, Drawing & Pointing Coordinates
Authenticate/ Profile & Progress
Web Application/ Documents
Images/ Coordinates
Student's PC or Smartphone
Other Students' and Teachers' PCs & Smartphones
Images/Coordinates (from WebRTC SDK)
Progress Data (from Authentication Server)
REC
Logging Server
Record data from each room

Figure 6. System configuration of integrated ShareHandy

camera functionality only and a PC for viewing and pointing out documents and images (S2 in Section 4).

We will add these three functions in Integrated ShareHandy. Figure 5 shows a prototype image of the Integrated ShareHandy. In the part of 1), users can share exercise documents. In the part of 2), each student's progress is displayed with a donut board over his/her icon as well as numerical value of percentage. This part also shows the name of the document his/her being viewed when the exercise uses multiple documents. In the part of 3) of Section B, the name of the user is displayed, which Authentication Server enables it.

C. *Implementation of added functions*

By implementing these additional functions, we made a few changes to the system configuration, shown in Figure 6.

Firebase was selected to authenticate users and share progress for additional functionality. Firebase are used as not only an Authentication Server but also a Realtime Database server, which stores data that is not handled in a real-time manner like user profiles and progress.

A web application through which students can access documents referred has been implemented.

A computer special for logging will connect to each room, which analyzes audio, video, point locations, and progresses to record users' status and operations when using ShareHandy. The logging system is designed as an independent program from ShareHandy. This is because WebRTC is basically used with Peer-to-Peer network not allowing a host server which can summarize the data for logging. These logs will be used to evaluate the effects and problems of the system on peer-teaching.

D. *Evaluation methods for system and peer-teaching*

We plan to evaluate Integrated ShareHandy in two steps.

The first step is to measure the System Usability Scale (SUS) score. SUS is a tool that is widely used for the evaluation of both hardware and software consideration and its scale measures ten questions, which answer with a five-point Likert scale. Through a questionnaire survey to assess whether the system is sufficient enough to conduct peer teaching, where the SUS score of the system must be more than seventy when the system is considered easy-to-use [11].

After confirming that the system clears the criteria, experiments will be conducted to apply this system to real exercises as the second step. In the step, how to evaluate the activation of peer teaching is important. We believe that it is necessary to compare cases where the improved system is used, where the previous system used and those where it is not used, and thereby to analyze how and how much peer-teaching has been promoted under a quantitative index to investigate the effectiveness of the system for peer-teaching.

To do so, we will use the system record, such as:

- audio and video,
- history of drawings and points,
- students' progresses, and
- the number and the time of teaching sessions

to analyze how peer-teaching took place and how the system effect on it. Also, conversation time other than peer-teaching can be an indicator to check the effect of reducing the emotional barriers.

E. Compare functions between some major systems

We compare main functions to use for exercise between major online meeting systems. Most of major systems like zoom and Microsoft Teams cannot point and draw to video without using screen-sharing, and no way to see the point where members are reading of document. One weakness of the proposed system is that teacher must upload documents to server, and it takes much work than to distribute documents by other system.

V. Conclusion

We proposed a peer-teaching support system for online exercises, ShareHandy, which can be used in the context of IoT prototyping where hardware assembly is handled. The system allows students to share videos of artifacts under development and point out areas of interest in real time using smartphones and PCs. Experiments to evaluate its usefulness in improving peer-teaching in online exercises had positive results, suggesting that the system is effective enough to support group work and peer-teaching. However, the frequency to use the system is lower than expected, and it is found that the system is not effective enough in promoting peer-teaching.

Based on the analysis, we planned to develop "Integrated ShareHandy" that comprehensively supports peer-teaching in online exercises, which supports sharing not only their physical artifacts visually but also referenced documents and students' progresses, enabling them to know who can help and work together to activate peer-teaching.

The future work includes to find an effective way to use the system in online exercises and to improve its usability not discouraging its use in pee-teaching. In addition, we will apply the system to more types and number of online exercises and will improve the system for effective peer-teaching.

References

[1] S. Sekiguchi and T. Nakajima, "ShareHandy: Peer Teaching Support System for Online Exercises of IoT Prototyping," CORETA 2021 - Advances on Core Technologies and Applications, Athens (Greece), November 14 - 18, 2021.

[2] S. Ramaswamy, I. Harris and U. Tschirner, "Student Peer Teaching: An Innovative Approach to Instruction in Science and Engineering Education," Journal of Science Education and Technology, vol. 10, no. 2, pp. 165-171, 2001.

[3] B. Goldschmid and M. L. Goldschmid, "Peer teaching in higher education: A review," Higher Education,vol. 5, pp. 9-33, 1976.

[4] S. Baddeley, "Online teaching: a reflection," The Journal of Classics Teaching, vol. 22, no. 44, pp. 109-116, 2021.

[5] M. C. Dorneich, B. O'Dwyer, A. R. Dolowitz, J. L. Styron and J. Grogan, "Application exercise design for team-based learning in online courses," New Directions for Teaching and Learning, vol. 2021, no. 165, pp. 41-52, 2021.

[6] S. Kitagami, K. Hasegawa, H. Koizumi and M. Inoue, "Online Training Environment for IoT Prototype Development", Proceedings of the Symposium on Information Education, pp. 240-243, 2020.

[7] J. O. Hamblen and G. M. E. V. Bekkum, "An Embedded Systems Laboratory to Support Rapid Prototyping of Robotics and the Internet of Things," IEEE Transaction on Education, vol. 56, no. 1, pp. 121-128, 2012.

[8] N. R. Dissanayake and K. A. Dias, "Web-based Applications: Extending the General Perspective of the Service of Web," 10th International Research Conference of KDU (KDU-IRC 2017) on Changing Dynamics in the Global Environment: Challenges and Opportunities, Aug. 2017.

[9] B. Sredojev, D. Samardzija and D. Posarac, "WebRTC technology overview and signaling solution design and implementation," 2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), pp. 1006-1009, 2015.

[10] ISO 9241-11: Ergonomics of human-system interaction, -Part 11: Usability: Definitions and concepts, 2018.

[11] A. Bangor, P. T. Kortum and J. T. Miller, "An Empirical Evaluation of the System Usability Scale," In International Journal of Human-Computer Interaction, Vol. 24, No. 6, pp. 574- 594, 2008.

An Intelligent Management System to Support Sustainable Urban Agriculture: the Case Study of the CIRC4FooD Platform

Dimitra Tsiakou, Lucyna Łękawska-Andrinopoulou, Marios Palazis-Aslanidis, Vassilis Nousis, Georgios Tsimiklis, Maria Krommyda, Angelos Amditis, Evangelia Latsa
Institute of Communication and Computer Systems, National Technical University of Athens
Athens, Greece
email: dimitra.tsiakou@iccs.gr

***Abstract*—The objective of this work is to improve the management of a vegetable garden through an Intelligent agriculture management system based on IoT, where the activities of a garden are handled through an online monitoring platform. The platform has been developed within the project 'A circular economy inspired food production system' (CIRC4FooD) as a part of the intelligent management system for the urban vegetable gardens, with the wider aim to engage the user and to promote and inform about circular economy and sustainable food systems. The CIRC4FooD integrated system consists of a garden equipped with a rainwater harvesting system, composting bin, and intelligent management system, namely the platform and incorporated Internet of Things (IoT) technologies: various sensors for soil and air parameters, rainwater tank and compost. The implementation of a dynamic rule engine consisting of three modules: i) environmental, ii) water tank, and iii) compost, allows the users to get notifications about any actions required or recommended from their side to keep the garden in proper condition, but also to assure environmental benefit (i.e., saving water resources by using harvested water, watering the garden only when the real need arises, use of compost as fertilizer, etc.). In this paper, the preliminary system and platform design are presented, which might be further improved, followed by the next steps that will be scheduled within the CIRC4FooD project to test the platform in real-life conditions and on different scales.**

Keywords-intelligent management system; platform; dynamic rule engine; urban agriculture; sustainability.

I. Introduction

In the original conference paper [1], about CIRC4FooD platform, we briefly introduced the project's benefits, focusing on the system's main functionalities. Thereafter the study aims to introduce in depth the additional developed features of the intelligent management system, highlighting the support for sustainable agriculture.

Food production poses a significant environmental burden that accounts for 10%-30% of an individual's environmental impact [2]. According a recent report from the Food and Agriculture Organization [3], current food production systems are facing several challenges: (a) growing demand for food, driven by increasing world population and urbanization; (b) diminishing land and water resources and their declining quality; (c) climate change, and at the same time significant contribution of the agricultural sector to this phenomenon; and (d) too few investments in solutions contributing to sustainable agriculture.

Due to the environmental stress on water bodies, harmful land use practices, soil depletion, and greenhouse gas emissions, the need for sustainable agriculture solutions is rapidly growing. The ever-present efforts to improve agricultural yield with fewer resources and labor have resulted in significant innovations throughout human history. Despite those efforts, the growing population rate never let the demand match the actual supply. Demand for food is growing at the same time the supply side faces constraints in land and farming inputs. The world population is expected to reach 9.8 billion in 2050, increasing approximately 25% from the current figure [4].

The listed challenges, in line with the destructive effect, that the traditional, linear economy has on the food system, pose increasing pressure on the food system, creating urgency for resilient and sustainable food systems.

CIRC4Food project introduces a solution for local food production systems, contributing to the following:

- the promotion and dissemination of a sustainable food system
- the facilitation of social interaction between residents
- the transformation of physical space
- the increase in awareness of healthy eating
- the education about affordable and fresh food production.

CIRC4Food system for food production consists of a vegetable garden equipped with a rainwater harvesting system and a composting bin. Many studies have proven the numerous environmental, social, and health benefits of urban vegetable gardens [5-7]. The CIRC4FooD system will be supported through an intelligent management system and will introduce a user-friendly CIRC4FooD platform with user-engaging functionalities to promote the sustainable use of natural resources.

The rest of the manuscript is structured as follows. Section II describes the smart farming concept and the integrated system for application in urban vegetable gardens proposed by the CIRC4FooD project. Section III describes the system design, including user requirements collection, as well as user journey and user flow, user types and the concept behind reward system to be developed. Section IV addresses the platform design including its architecture, the rule engine and the platform view. Finally, Section V concludes on the work and outlines next steps. The work is closed by acknowledgements.

II. Smart Farming and CIRC4FooD

The introduction of technology in agriculture aims to an intensive increase in food productivity as well as removing any apprehensions concerning the scarcity of food in the future. The technological applications related to agricultural aspects were classified into three categories, namely data sources and collection, machine learning (ML) methodologies for agricultural data, and intelligent knowledge acquisition. Considering the imperative need for action in the promotion of sustainable food systems, in Figure 1 are highlighted the key drivers of technology in the agriculture sector.

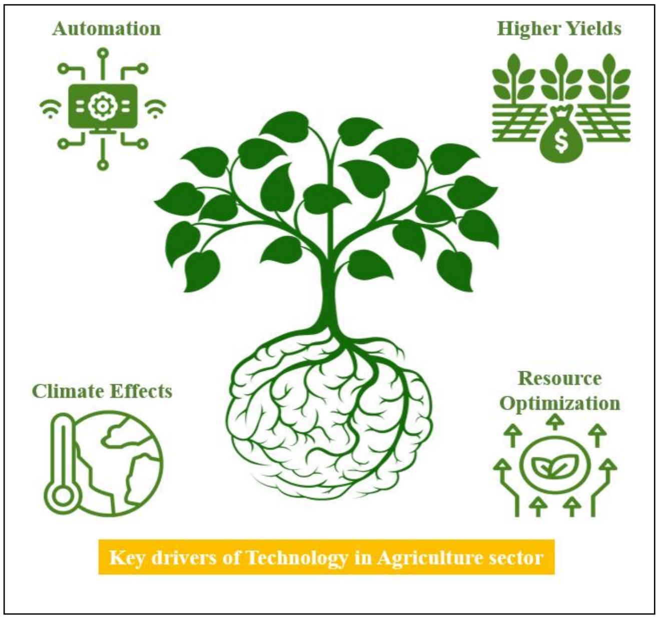


Figure 1. Key drivers of Technology in Agriculture sector.

Smart farming, as a coupling of information, communication, and control technologies in agriculture, is an idea that gains ground gradually. Smart farming is a management concept of using modern technology to increase the quantity and quality of agricultural products. The concept involves an integration of information and communication technologies into machines, sensors, actuators, and network equipment for use in agricultural production systems [8].

These applications are the driving force for the development of innovation in precision and sustainable farming. Although, there are some notable barriers regarding the technology implementation in smart farming, presented in Figure 2.

In a smart farming scenario, large amounts of real-time and high-resolution data are generated from remote and automated sensor systems. The data can represent different aspects of farming, including but not limited to livestock, crops, soil, and the environment [9]. Numerous software developments are available in the market [10-11], designed to make the farmer's tasks more efficient. However, as every particular application has specific characteristics, no "one-size fits all" technology is available. To improve sustainability, there is a need for site-specific strategies for both decision-makers and farmers. Some of the specific aspects of interest of the system design and selection involve the assessment of the techno-economic and environmental impact of an urban farming system, the choice of crops and the optimization of economic and environmental parameters. The optimal design and operational plan pose a significant challenge [12]. Nevertheless, the usage of all these information in the field is usually limited to the aforementioned aspects, and for this reason, an inclusive and multipurpose monitoring platform is proposed, which explicitly supports the management and optimization of the performance of a vegetable garden, with special attention to compost fertilizers production and use and operation of water harvesting system contributing to sustainable urban farming systems.

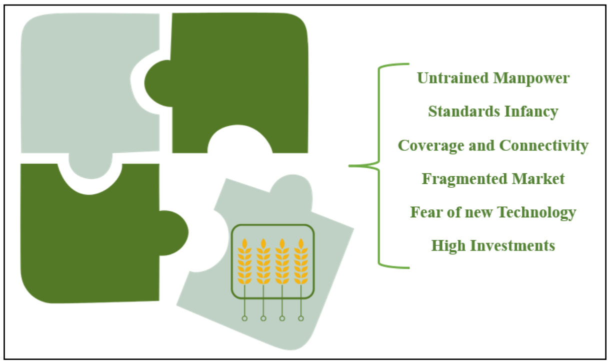


Figure 2. Notable barriers of Technology implementation in smart farming.

Most of the proposed platforms in smart farming have focused on a specific aspect of smart farming, such as crop production recommendations [13], big data technologies, data transformation [14], reliability [15], and business models [16]. In this paper, we propose the CIRC4FooD platform approach as a unified solution that facilitates smart farming applications following the sustainable use of resources. Additionally, this approach imposes learning techniques for urban farming and provides data to track crop cycle information, fertilizer, and water in a secure manner for their decisions and data management.

In regards to that, the intelligent urban vegetable gardens in CIRC4FooD project, use technological resources that help in various stages of the production process, such as monitoring of crops, irrigation and composting process control. More precise, CIRC4FooD integrated system, as revealed in Figure 3, for implementation in urban vegetable gardens consists of the following elements: i) the garden itself, ii) rainwater harvesting and storing system, iii) composting bin, iv) intelligent management system integrating on-line monitoring platform and IoT technologies (including sensors).

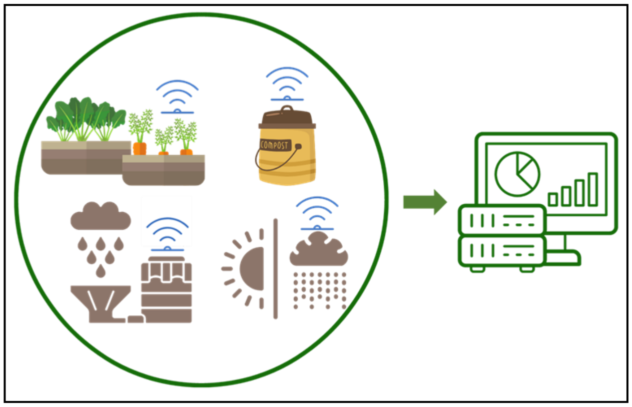


Figure 3. CIRC4FooD integrated system.

The aforementioned sensors will be acquired within the purpose of the project, and upon completion, the participating users will retain those sensors. More specifically, soil moisture sensors will be installed in the vegetable garden, humidity and temperature sensors inside the compost bin, as well as water level and Total Dissolved Solids (TDS) sensors inside the water tank.

III. System Design

A. *User Requirements Collection and KPIs clarification*

The purpose of the requirements collection is to understand the needs of the end-users and the problems they seek to resolve with the specific platform.

The process of gathering the requirements of the system followed the subsequent four main stages:

- Elicitation: In this stage, the project team collected the requirements of the end-users
- Analysis/Processing (Analysis): In this stage, the project team tried to understand the requirements of the end users and model the desired operation of the system based on their requirements.
- Specification: In this stage, the project team tried to document the functionality of the proposed software system.
- Validation: In this stage, the project team checked that the system specifications match the initial requirements of the end users.

All involved/interested members jointly came to a decision on what the final system requirements/specifications will be. For the CIRC4FooD system, the stakeholders involved in gathering the requirements were:

- Domain experts: People who have been in the field for years and could contribute to the requirements/specifications of the CIRC4FooD system since they know very well the needs that the system will cover, as well as the risks that are or may arise during the pilot period. The experts in the area are made up of experts/professionals on the cultivation, employees of the green service of the Municipality, and experts on the specific technology.
- End users: People or groups of people who will use the final product and will be able to evaluate it. The end users are the owners of non-private vegetable gardens who wish to use the CIRC4FooD system.
- Software Engineers: Engineers who will develop the system and will be able to train the end users on the innovations of the product, both in terms of hardware and software technology. The software engineers, in this case, are from the CIRC4FooD project team.

After the consideration of the aforementioned, in order to extract the requirements from the end-user's standpoint, the following factors affecting urban farming were taken into consideration:

- *Weather*: Farming mainly depends on weather conditions. Farmers face great risk in growing crops, as insufficient rainfall and water supply can damage the crop or lead to a decrease in farm produce. Considering the fact that different plants require different parameters of weather conditions, an all-purpose and simplified system – that is suitable for numerous crops – was established.
- *Lack of knowledge and skill*: Literacy is one of the most important factors affecting all the sectors. Lack of literacy results in farmers being unaware of changes occurring in the farming sector. Informing end-users about the dedicated activities regarding the vegetable garden motivates the interested parties to sustainable thinking.
- *Seeds/fertilizers/disease*: To grow crops of good quality, selection of seeds and appropriate knowledge about fertilizers are required. Additionally, timely and proper detection of plants affected by disease can save the farmer from loss and helps in gaining crop security. A repository with information about plants, their characteristics, possible diseases, and advice on handling them will be available for users.
- *Water scarcity:* A more efficient irrigation management focused on reducing the capacities of water applied and therefore optimizing the conservation of irrigation water, is conceivable through the platform which helps the end-user to plan the irrigation activities.
- *Promotion of circularity in food production:* The shift from a linear model to a circular model can meaningfully decrease the negative burdens on the environment and contribute to reestablishing biodiversity and natural resources. With this aim, the presented platform can play a relevant role in setting the paths of this transition, nurturing the shift towards a more sustainable agriculture.

Based on the above-mentioned factors and through a dedicated questionnaire that was conducted during the CIRC4FooD project by e-Trikala, which is the responsible partner for the implementation of the urban vegetable gardens in the city of Trikala during the CIRC4FooD project, user requirements were collected. With the completion of the user requirements, as it emerged from filling out the questionnaire, the analysis/processing of the results followed, where the next step was to better understand the user requirements and to model the desired operation of the system based on the gathered requirements.

Moreover, in addition to the requirements that focus on factors affecting urban agriculture, called standardized data (sensor data, weather data, user data, or coded data), another set of requirements was selected, which focus on functional requirements. Finally, the users, beyond the functional requirements that they wish the system to perform, also have expectations for how it should work. The characteristics that fall into this category are how easy a system is to use, how fast it is, how often it fails and how it will be able to handle unforeseen conditions. The above are features or quality factors of the software and are part of the non-functional system requirements. These characteristics are difficult to define, but in order to perceive the success or failure of the systems, compliance with the non-functional requirements plays an important role. Therefore, while eliciting the requirements, the quality expectations of the users also were taken into consideration.

Functional requirements describe the functional capabilities or services of the system and depend on the type of software, the expected users, and on the type of system in which the software is used. On the other hand, non-functional requirements describe system properties that are usually expressed based on form characteristics: Performance, Usability, Security, Legislative and Privacy. In other words, they describe how (or how well) the system will support the functional requirements and are considered "constraints" that limit the ways in which users can realize the functional requirements. Some of the functional and non-functional requirements used to finalize the user requirements are as follows:

- Personalization
- Authentication/Security
- Authorization
- Scalability
- Reusability
- Usability
- Performance
- Localization

Having concluded on user requirements, Key Performance Indicators (KPIs) were clarified, which will measure the performance of the pilot applications of the project, as well as the indicators that will be able to identify the social acceptance and social impact of CIRC4FooD. The methodology followed is, on the one hand, based on the initial objectives set when designing the food production system inspired by the CIRC4FooD circular economy, but also on its expected results and on the other hand, on the design of the system and the characteristics of the pilots.

In addition to the quantitative objectives related to the performance of the system, the environmental, economic, and social benefits expected as a whole from the implementation of CIRC4FooD, but also from its specific applications at three scales within the city of Trikala, were taken into account. The methodology followed is described in Figure 4.

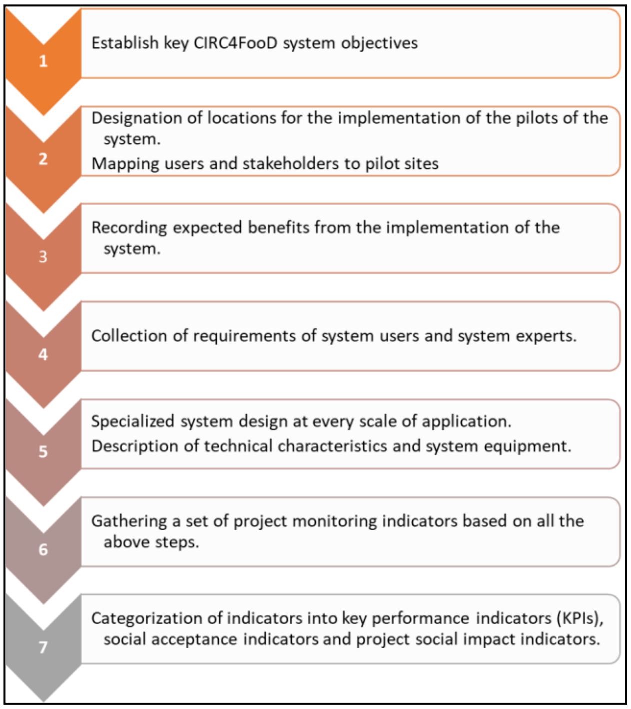

1
Establish key CIRC4FooD system objectives
2
Designation of locations for the implementation of the pilots of the system.
Mapping users and stakeholders to pilot sites
3
Recording expected benefits from the implementation of the system.
4
Collection of requirements of system users and system experts.
5
Specialized system design at every scale of application.
Description of technical characteristics and system equipment.
6
Gathering a set of project monitoring indicators based on all the above steps.
7
Categorization of indicators into key performance indicators (KPIs), social acceptance indicators and project social impact indicators.

Figure 4. CIRC4FooD indicator selection methodology.

The user journey and flow were considered to guarantee that the interaction process between the users and the system will be effective and are analysed in the following section. While both tools are used to communicate the design of the system through the lens of the users' goals, they aren't synonyms because they focus on different aspects of the created system.

B. *User Journey and User Flow*

User Journey refers to the scenarios in which the user interacts with the product. This visual representation is commonly created as a timeline of actions or steps by a facilitator, built up on feedback collected methodically (via observations, interviews, focus groups, etc.). As a result, the technique's main function is to assume and demonstrate the current and possible way in which the user can interact with the process. User Journeys deal with the emotions, pain points, and motivations of the end-users [17]. For this aim, the establishment started with the completion of a user journey map. The developed map is a visualization of the step-by-step experience as the user goes through the platform, following the diagram as shown in Figures 5 and 6 for climate and sustainability activities.

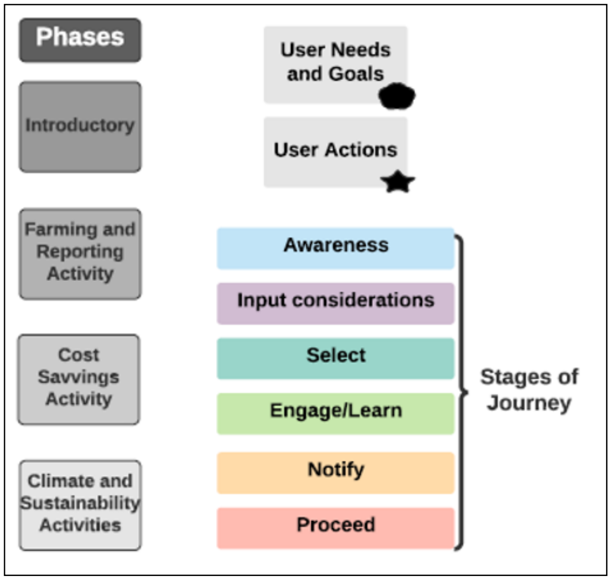
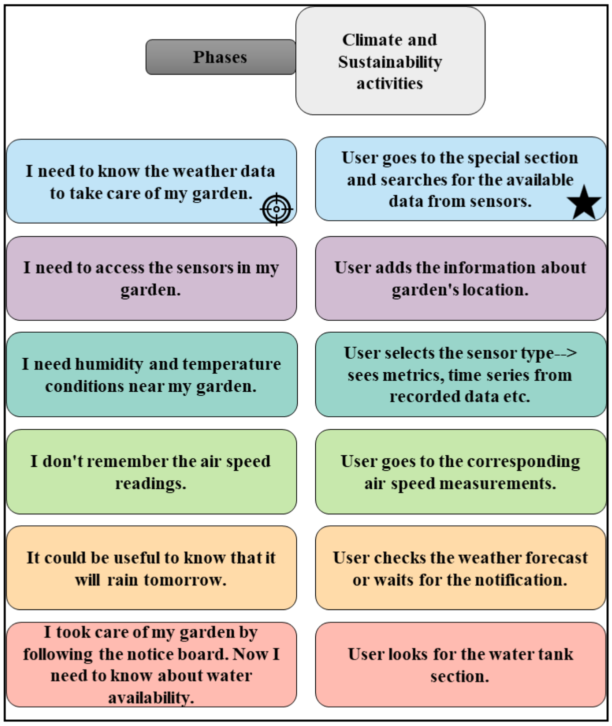


Figure 5. Concept of User Journey.



I need to know the weather data to take care of my garden.	User goes to the special section and searches for the available data from sensors.
I need to access the sensors in my garden.	User adds the information about garden's location.
I need humidity and temperature conditions near my garden.	User selects the sensor type--> sees metrics, time series from recorded data etc.
I don't remember the air speed readings.	User goes to the corresponding air speed measurements.
It could be useful to know that it will rain tomorrow.	User checks the weather forecast or waits for the notification.
I took care of my garden by following the notice board. Now I need to know about water availability.	User looks for the water tank section.

Figure 6. User Journey for climate and sustainability activities.

In order to put the users' needs and wants at the forefront of this design process, a guaranteed way to achieve this is via user flows. User Flow refers to the process in which user takes advantage of the complex routes through a series of templates designed for a product to accomplish their goal. It is created to predict and show the possible routes through which the user interacts with the product. User Flows are usually depicted by flow charts, and they are a set of steps taken by a user to achieve a goal using a digital product. Rather than demonstrating how the users are supposed to feel, a User Flow is the breakdown of the actual user interface. Designing how a user interacts with a product is a key step in figuring out where the issues may arise in task flows [18].

Having finalized the User Journey and the User Flow, user types were selected, and ways to accomplish user engagement with maximum impact were identified in order to understand users, challenge assumptions, redefine problems and create innovative solutions to prototype and test.

C. *User Types and Reward System*

To assure comprehensive and effective knowledge acquisition, users of the platform are assigned into one of the following user types: novice, advanced beginner, competent and experienced. The assignment is based on the self-assessment performed by the user based on the information provided for each level of expertise and adapted from [19] for the needs of the platform. Nevertheless, the platform is able to accommodate a variety of user types besides the intended core set of users.

After identifying the core qualities of the target users, the benefits of a reward system in any working environment were explored [20]:

- It boosts the user base and encourages more users to get on board with the system.
- It improves confidence and esteem towards the system.
- It makes the users share this concept with others.
- It helps the users develop a trust factor for the system when they see some positive benefits of the system.

A reward system will be implemented during the user training phase and will be based on user engagement, allowing the users that are most active to be upgraded to higher levels of expertise as they gain knowledge and know-how. One of the excessive aspects of rewards is that they gather much larger sets of data. Simple rewards may incorporate a few data elements. More sophisticated rewarding rules may aggregate scores from multiple more attentive rewarding rule sets and integrate important supplementary metrics. With the capability of rewarding rules to summarize data, decisions and actions can be made much faster.

CIRC4FooD platform follows a method of reward with points and scores, which are the most common type of reward. Points and scores were the results of a change in behavior. Aiming to support sustainable urban agriculture through this intelligent management system, the points of the reward system are inextricably linked with the rule engine of the platform (described below in Section Dynamic Rule Engine). Furthermore, some points are awarded based on some of the users' actions that are directly related to the work in their vegetable garden, and some points are related to more simplified actions (e.g., rate of visiting the notification page, signing in, etc.).

The role of a reward aligned with personal values may serve as a driver of integrated sustainability, thereby increasing motivation to apply conservation practices over time. To increase the likelihood that motivation is maintained or enhanced, CIRC4FooD recommends that specific values of rewards should be explored in future interventions after the user training phase. A survey will be distributed to participators to determine whether the positive motivation for the use of the platform predicts the scale for satisfaction with urban farming and sustainable practices in agriculture. This survey will contribute to relevant issues by identifying factors that could be improved to enhance learning and adaptation to circular economy practices.

IV. Platform Design

A. *Architecture*

The architecture of the web platform follows the principles of a MERN full-stack development. MERN stack is a JavaScript stack, that is used for easier and faster deployment of full-stack web applications. MERN Stack comprises of 4 technologies namely: MongoDB, Express, React and Node.js and it is designed to make the development process smoother and easier as depicted in Figure 7.

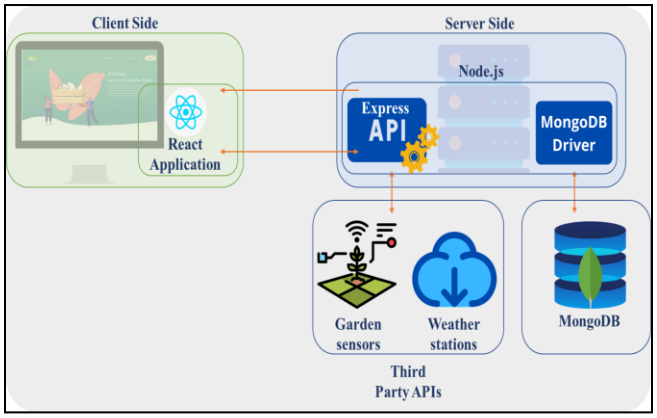


Figure 7. Architecture of the CIRC4FooD platform (own work).

MongoDB is an open-source document database built on a horizontal scale-out architecture that uses a flexible schema for storing data [21]. In the CIRC4FooD platform, MongoDB is used to store as object-collections information related to the system users, their integrated system and the notifications extracted from the rule-engine.

Express is a prebuilt Node.js framework that can help creating server-side web applications and APIs faster and smarter. Simplicity, minimalism, flexibility and scalability are some of its characteristics, and since it is made in Node.js itself, it inherited its performance as well.

On the client-side part there is React.js. React is an open-source, component-based front-end library, responsible only for the view layer of the application. It is maintained by Facebook. Using functional programming, hooks and JSX, React designs simple views for each state in the application, and will efficiently update and render just the right component when the data changes. As a front-end framework, React communicates with the back-end by making API calls on the endpoints created by Express.

The last technology is Node.js. Node is an open-source development platform for executing JavaScript code server-side. It is intended to run on a dedicated HTTP server and to employ a single thread with one process at a time [22]. Node.js applications are event-based and run asynchronously. In the CIRC4FooD platform, Node.js has a central role as it is the one responsible for serving the client-side code, managing the Express APIs and communicating with the MongoDB database. Moreover, Node.js fetches data from a third-party API related to the current values of each sensor placed on the gardens and stores them to the proper collections in the Mongo database. Furthermore, it handles the authentication and role system for the users as long as the reward system and the dynamic rule-engine, that creates the proper notifications for the platform users. More about the rule-engine and the reward system will be presented in the following paragraphs.

B. *Dynamic Rule Engine*

The first step to develop our intelligent management platform was to know the variable inputs the system has and how they have to be handled by using a set of rules, aiming to enhance the sustainability and competitiveness of the activities taking place. The data from the garden sensors and weather stations, once pre-processed and formatted, are sent to the rule engine for rule-based processing to produce relevant outcomes. Historical database information may be required for some rules.

The process of preparing a rule base in CIRC4FooD can be divided into several consecutive steps that are presented in Figure 8 below, wherein several layers are created: data collection from several sensors (as described in Figure 3), system modelling and rule selection, environmental sustainability, deployment of rules and also system optimization. The rules guiding the dynamic rule engine were constructed to increase the productivity, effectiveness, and performance of connected sensors, with two primary purposes: i) to promote sustainability and natural resource preservation and ii) to maximize user engagement.

CIRC4FooD solution helps the end users in smart farming operations in a more user-friendly format, informing the latter about the circular economy and guiding them towards increasing productivity and reducing resource wastage. Additionally, this approach increases food security, production, and also sustainability by providing data from the installed sensors during the whole process of farming. Moreover, this solution provides an exceptional prospect for building management systems by combining data from varied sources.

In order to assure user engagement, the number of received notifications, but also their exact content, is adjusted to the level of experience of the user. The higher the level of experience, the higher number of notifications the user gets, while their content becomes less explanatory and more informative. This is illustrated well in Figure 8 and with the examples of notifications presented in Table 1.

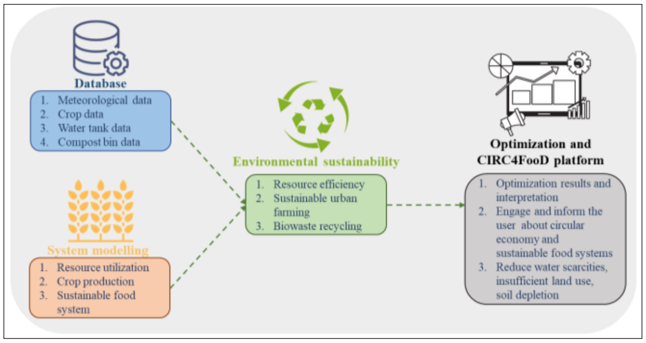

Database
1. Meteorological data
2. Crop data
3. Water tank data
4. Compost bin data
System modelling
1. Resource utilization
2. Crop production
3. Sustainable food system
Environmental sustainability
1. Resource efficiency
2. Sustainable urban farming
3. Biowaste recycling
Optimization and CIRC4FooD platform
1. Optimization results and interpretation
2. Engage and inform the user about circular economy and sustainable food systems
3. Reduce water scarcities, insufficient land use, soil depletion

Figure 8. Outline of the process of prepearation of the rule base in CIRC4FooD.

TABLE I. EXAMPLES OF NOTIFICATIONS FOR FOUR USER TYPES

Condition	**User Type**			
	Novice	***Advanced Beginner***	***Competent***	***Experienced***
Air humidity < 30% and soil moisture <20%	The soil moisture levels are below optimal and air humidity is low. Consider watering your crops.	Crops in conditions like today's will most probably need water.	Air humidity today will be low and soil moisture readings indicate that your crops need water.	Air humidity < 30% and soil moisture <20%
Water tank level: full	N/A	Your water tank is full, now is the time to save natural water resources.	Water tank is full.	Water tank is full.
Compost moisture >60 %	Your pile moisture levels are above the optimum. To keep them within the optimal range mix in some newspaper strips, dry wood chips or pieces of cardboard.	Your compost is too wet. Mix in newspaper strips, dry wood chips or pieces of cardboard.	The optimal levels of moisture range between 40 to 60 %. Today the moisture level is >60 %. Take appropriate actions!	Compost moisture > 60 %
Probability of precipitation >60 %	Today most probably it is going to rain.	Today most probably it is going to rain.	Today most probably it is going to rain.	Probability of Precipitation >60 %
Probability of precipitation >60%, soil moisture < 20 %	The soil moisture of your crops is low and today there is a high probability of precipitation. Wait for the rain to water your crops and save water.	The soil moisture of your crops is below 20% and today there is a high probability of precipitation. Wait for the rain to water your crops and save water.	Soil moisture < 20 %, but rain is expected! Wait with watering your garden, and water it only if the weather forecast proves wrong and there is no rain.	probability of precipitation >60%, soil moisture < 20 %
Compost temperature <25°C	Have an eye on your compost, the temperature is too low, the process is not taking place! Action necessary.	Have an eye on you compost, the temperature is too low, the process is not taking place! Action necessary.	Compost Temperature <25°C	Compost Temperature <25°C
TDS >2000ppm	If water uptake is appreciably reduced, the plant slows its rate of growth. A TDS level above 2000 is completely unsafe and dangerous for any use.	If water uptake is appreciably reduced, the plant slows its rate of growth. A TDS level above 2000 is completely unsafe and dangerous for any use.	The TDS level inside your water tank is unsafe for any use. Consider not to use this water for your activities.	TDS > 2000ppm

The rule engine consists of three modules: i) environment, ii) water tank, and iii) compost. The module environment gathers rules related to the environmental parameters influencing plant growth, with special attention to those related to watering. Parameters taken into consideration in this module are the following: soil moisture, air humidity, the likelihood of precipitation, temperature, light intensity, and wind speed.

Module water tank describes the rules associated with water level and TDS amount in the water tank, but also the water level in combination with the probability of precipitation and soil moisture, to assure that water is used only when a need arises.

Module compost is built to support the composting bin notifications in relation to compost temperature and compost moisture. Compost temperature and compost moisture are two critical parameters for the process of compost production. Compost temperature is additionally a crucial parameter in compost monitoring [23]. Monitoring temperature evolution over time provides critical information about the course of composting and assures the safety of the produced material (eliminating the risk of microbial contamination) [24], as well as the safety of the process itself (avoiding fire hazards).

The conditions are assigned impact values based on the literature research and the CIRC4Food aspirations, namely user engagement, awareness raising, and saving natural resources, especially water. The impact values affect the notifications scheme of the rule engine, which is built on threshold values. The user receives a notification only if the threshold value is equal to or surpasses the impact value. The threshold value (and, therefore, whether or not the notification will be received) is determined by the user type.

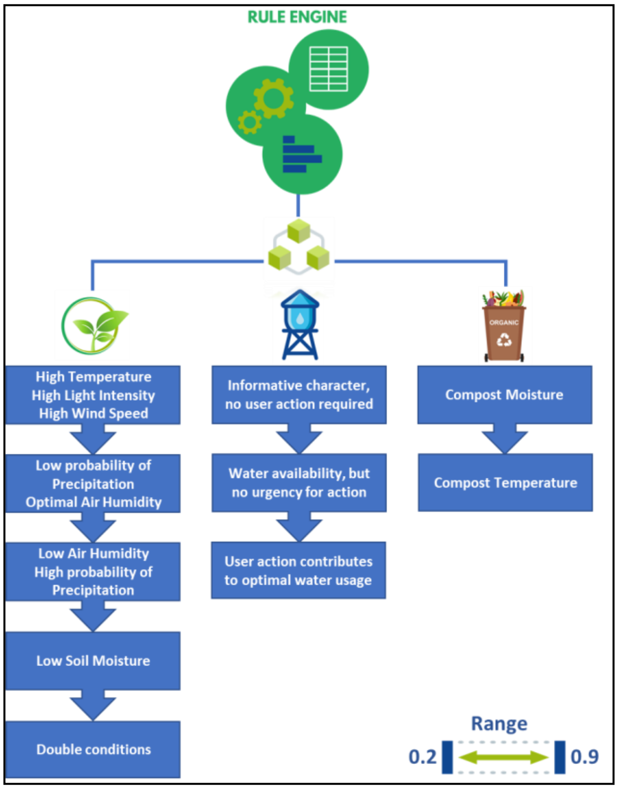

RULE ENGINE
High Temperature
High Light Intensity
High Wind Speed
Low probability of Precipitation
Optimal Air Humidity
Low Air Humidity
High probability of Precipitation
Low Soil Moisture
Double conditions
Informative character, no user action required
Water availability, but no urgency for action
User action contributes to optimal water usage
ORGANIC
Compost Moisture
Compost Temperature
Range
0.2
0.9

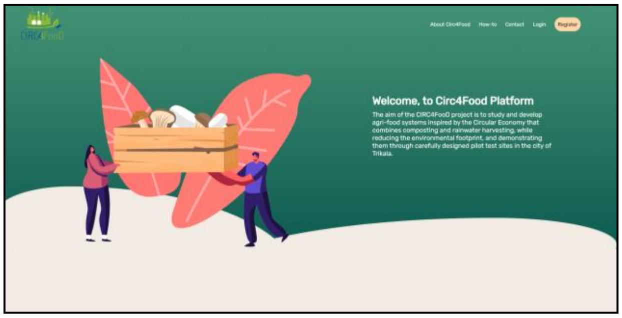
Figure 9. Dynamic rule engine for the three modules.

The dynamic rule engine is designed in such way that additional rules can be implemented if the need arises. The followed procedure for the implementation of a dynamic rule engine for the three modules is defined in Figure 9.

C. *CIRC4FooD Platform view*

The last step to make the management platform usable was to develop a user interface (UI) to simplify its use by dedicated users. Additionally, CIRC4FooD should be tested in terms of data collection and operation. In this section, the platform view development is presented, with the aim of offering from top to bottom usability level for non-IT-experts. CIRC4FooD platform integrates a web-based data-mining system (third-party APIs). The platform provides an important interaction model for the smart farming sensors by letting users acquire information related to the integrated garden sensors.

The front-end application exposes to the user a pleasant and interactive interface through which the user can access all the services exposed by the back-end application. The CIRC4FooD interface is responsive, so the user experience on the platform is of high quality regardless of the device used (laptop, PC, tablet, or phone). In addition, the CIRC4FooD platform allows end-users to explore and analyse agricultural and weather data with zero-programming efforts. The initial UI is depicted in Figures 10-16.



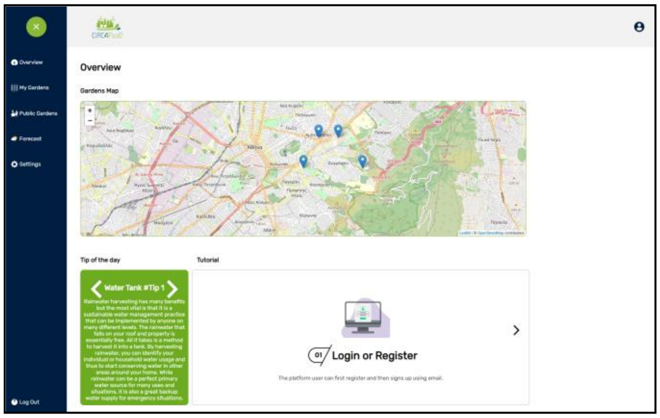
Welcome, to Circ4Food Platform

The aim of the CIRC4FooD project is to study and develop agri-food systems inspired by the Circular Economy that combines composting and rainwater harvesting, while reducing the environmental footprint, and demonstrating them through carefully designed pilot test sites in the city of Trikala.

Figure 10. Welcome Page of the CIRC4FooD Platform.

Figure 10 shows the welcome page of the platform, where the user can read a few words about the project within the platform that was created, as well as log in if they are already a registered user, otherwise create their profile. For usual registration, the information that must be filled in is as follows:

- Personal data: first name, last name, email address (used for sending activation email and other platform-specific emails);
- Information used by the CIRC4FooD platform: username, password (both used for authentication), and preferred language (the default, for now, is English).



Overview

Gardens Map

Rainwater harvesting has many benefits but the most vital is that it is a sustainable water management practice that can be implemented by anyone on many different levels. The rainwater that falls on your roof and property is essentially free. All it takes is a method to harvest it into a tank. By harvesting rainwater, you can identify your individual or household water usage and thus to start conserving water in other areas around your home. While rainwater can be a perfect primary water source for many uses and situations, it is also a great backup water supply for emergency situations.

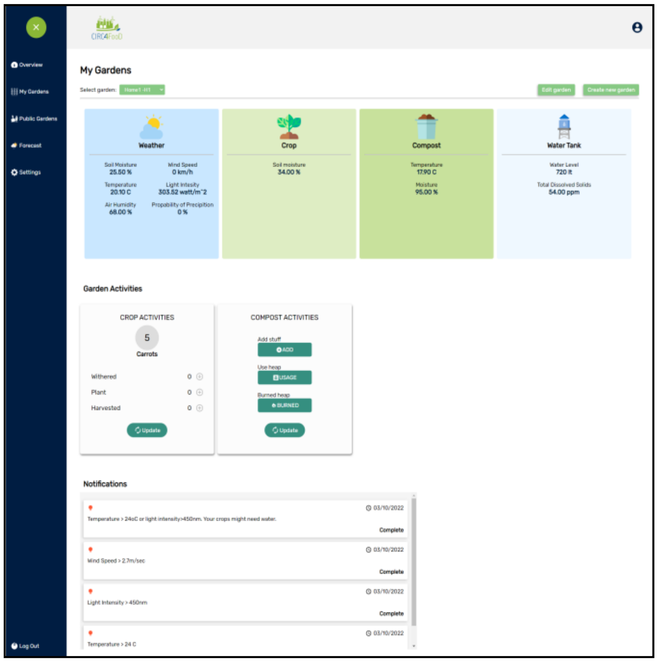
The platform user can first register and then signs up using email.

Figure 11. Homepage/Dashboard view.

When the user is logged in, the platform looks like a dashboard. The homepage contains a map which is resembling with google maps, with the difference that the location of public vegetable gardens that have been created in the city of Trikala is displayed, and the gardens owned by the current user are represented with a bold point. In Figure 11, there is the homepage of the platform. On this particular page, the user can initially be familiar with the tutorial, where useful instructions guide the user in using the platform. The user also has access to the tip of the day section and has the chance to explore useful information related to urban agriculture and the individual systems installed in the garden.

Figure 12 reflects the 'My garden' page. Here the user comes into first contact with the measurements of the sensors installed in the garden, compost, and water collection tank. Also, the user can see some of the measurements of the meteorological stations located near the garden, which are considered important for agricultural processes. Additionally, the user will have the opportunity to update the system about the actions taken regarding the three subsystems (water tank, garden, compost), as well as to see all the notifications that have been produced by the dynamic rule engine. Within the 'My gardens' page, the user can create a new garden or edit the already existing garden and must provide advanced data about the garden. The process of creating a new garden is clear in Figure 13, where the user follows five steps:

1. garden title
2. crop information
3. input of sensor IDs to connect to own garden
4. garden locations
5. declare the gardem as public garden or not.



My Gardens

Select garden: Home1 -H1

Weather

Soil Moisture 25.50 %
Wind Speed 0 km/h
Temperature 20.10 C
Light Intesity 303.52 watt/m^2
Air Humidity 68.00 %
Propability of Precipition 0 %

Crop

Soil moisture 34.00 %

Compost

Temperature 17.90 C
Moisture 95.00 %

Water Tank

Water Level 720 lt
Total Dissolved Solids 54.00 ppm

Garden Activities

CROP ACTIVITIES

5 Carrots

Withered	0
Plant	0
Harvested	0

COMPOST ACTIVITIES

Add stuff

Use heap

Burned heap

Notifications

03/10/2022 Temperature > 24oC or light intensity>450nm. Your crops might need water. Complete

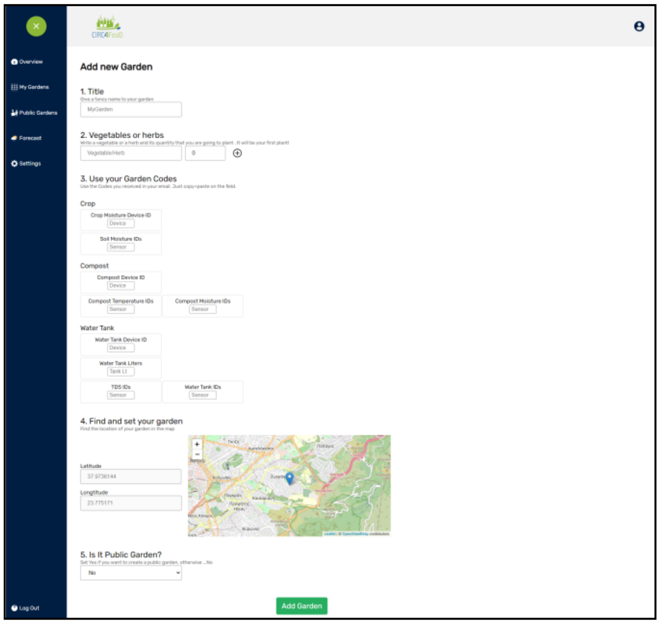
03/10/2022 Wind Speed > 2.7m/sec Complete

03/10/2022 Light Intensity > 450nm Complete

03/10/2022 Temperature > 24 C

Figure 12. My gardens page.

In Figure 14, there is the 'Public gardens' page, in which the user has access only in view mode to the different public vegetable gardens in the city of Trikala. The purpose of this page is to inform the user about the development of other gardens and to educate them about the processes of urban agriculture through general information about other gardens as best practices.



Add new Garden

1. Title
Give a fancy name to your garden
MyGarden

2. Vegetables or herbs
Write a vegetable or a herb and its quantity that you are going to plant . It will be your first plant!
Vegetable/Herb 0

3. Use your Garden Codes
Use the Codes you received in your email. Just copy+paste on the field.

Crop
Crop Moisture Device ID: Device
Soil Moisture IDs: Sensor

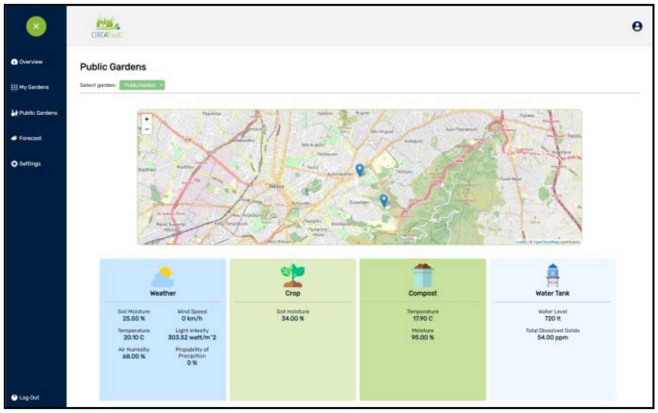
Compost
Compost Device ID: Device
Compost Temperature IDs: Sensor
Compost Moisture IDs: Sensor

Water Tank
Water Tank Device ID: Device
Water Tank Liters: Tank Lt
TDS IDs: Sensor
Water Tank IDs: Sensor

4. Find and set your garden
Find the location of your garden in the map
Latitude: 37.9736144
Longtitude: 23.775171

5. Is It Public Garden?
Set Yes if you want to create a public garden, otherwise ...No
No

Figure 13. Add new garden page.



Public Gardens

Select garden: PublicGarden

Weather

Soil Moisture 25.50 %
Wind Speed 0 km/h
Temperature 20.10 C
Light Intesity 303.52 watt/m^2
Air Humidity 68.00 %
Propability of Precipition 0 %

Crop

Soil moisture 34.00 %

Compost

Temperature 17.90 C
Moisture 95.00 %

Water Tank

Water Level 720 lt
Total Dissolved Solids 54.00 ppm

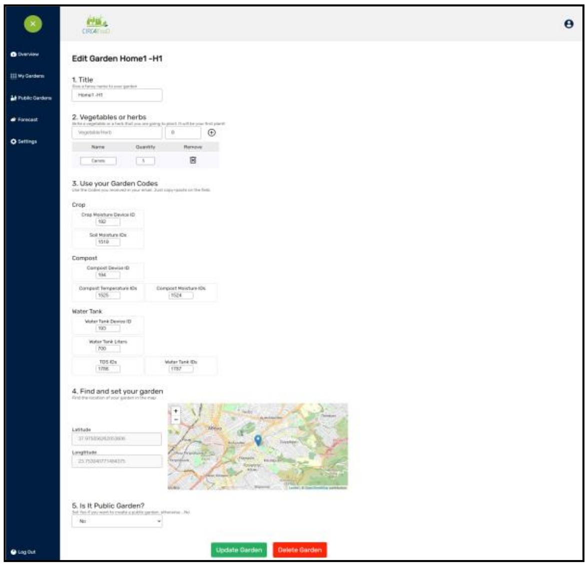
Figure 14. Public gardens page.

In Figure 15, the user can edit the existing garden, change the imported information and then update the new information about their own garden. The information regarding the sensors is provided to the user upon installation in order to make the process easier. As mentioned before, all information about the sensors comes from third-party APIs within the project, so the process of creating gardens is a bit more complicated. The goal is, after the project is over, to simplify this process as much as possible and make it more user-friendly and engaging.

In Figure 16, the weather forecast is depicted alongside weather service, in which the user can view various information about the conditions regarding each garden individually, based on its location. The user can choose through a drop-down menu the desired weather station. For each selected station, the CIRC4FooD platform displays the basic weather information (temperature, humidity, daily rainfall, hourly rainfall, wind speed, wind direction, light intensity, UV power, and soil moisture). Regarding weather forecast, the information is taken from an embedded weather widget, and the provider is called windy.com. This provider is open and can be used by anyone. Also, the platform has used some parameters to simplify the use of this widget, like the latitude and longitude of the city of Trikala and the zoom feature (value 11) to create a closer map. Moreover, this widget gives a ten days period of time forecast, and a variety of information are displayed: temperature, clouds, CO and SO_2 concertation, pressure, wind information, rainfall, real-time lightning strikes, weather radar, and much more parameters.

Some of the used units are:

- Celsius degrees (°C) for temperature;
- Cloud color, shape, and size for cloud indicator;
- Parts Per Billion by Volume (PPBV) for CO concertation indicator
- kg/m^2 for Sulfur dioxide (SO_2)
- Millimeter for rainfall;
- Knot (kt) for wind speed and wind gusts;
- Wind direction using the flag as an indication, etc.



Edit Garden Home1 -H1

1. Title

Home1 -H1

2. Vegetables or herbs

Name	Quantity	Remove
Carrots	5	☒

3. Use your Garden Codes

Crop

Crop Moisture Device ID: 192

Soil Moisture IDs: 1519

Compost

Compost Device ID: 194

Compost Temperature IDs: 1525

Compost Moisture IDs: 1524

Water Tank

Water Tank Device ID: 193

Water Tank Liters: 700

TDS IDs: 1786

Water Tank IDs: 1787

4. Find and set your garden

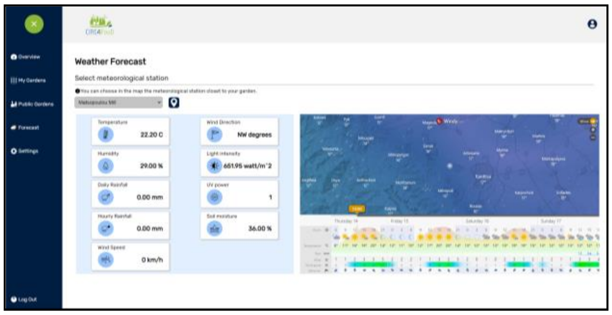
Latitude

Longitude

5. Is It Public Garden?

No

Figure 15. Edit garden page.



Weather Forecast

Select meteorological station

You can choose in the map the meteorological station closest to your garden.

Temperature 22.20 C

Humidity 29.00 %

Daily Rainfall 0.00 mm

Hourly Rainfall 0.00 mm

Wind Speed 0 km/h

Wind Direction NW degrees

Light intensity 651.95 watt/m^2

UV power 1

Soil moisture 36.00 %

Figure 16. Weather forecast page.

The User Interface (UI) exploits concepts that the user is familiar with and facilitates further understanding (through, e.g., data diagrams, sensor devices, tutorial, etc.), as defined in the data model and the rule engine.

The UI is still a work in progress, aiming to achieve an intuitive visualization that will depict the requirements of the end-users in order to contribute to the promotion of sustainable urban agriculture. For this specific purpose, the CIRC4FooD platform is intended to integrate, in the next version, the following five key elements:

First, it will be useful for the user to click on a garden and be redirected to the garden visualization page of the clicked garden (public and private). On the right side of the platform, the user will see their profile picture (by clicking on it will have access to the profile page) along with a drop-down menu where they will access all the components offered by the platform or can go to the view page of any of their registered gardens. On its profile page, the user will have the opportunity to review any activity on the platform, like viewing registered gardens, the tip of the day, tutorial, weather forecast, etc. When watching other gardens, the number of details displayed is in accordance with the privacy options selected during garden creation. CIRC4FooD is intended to give the user the opportunity of sending an email to the users of public gardens and be informed about various procedures regarding their farming activities, with the main purpose of training and educating themselves.

Second, there will be, at the top, a navigation bar through which the user will be able to search for private gardens and public ones. The user will also be able to find a new message indicator and a notification indicator. Clicking on a specific notification will open a dropdown menu with all the notifications, while clicking on the new message indicator will open the new notifications that are not yet addressed by the user. Also, from the top bar, the platform language will be able to change. After authentication, the default language is the one chosen by the user while creating the account. For the moment, the available option is English, but in the following version, the user will have the chance to choose between Greek and English.

Third, in the current version, in order to register into the platform, the user must fill in and submit a form from the registration page. They will also have the option to use a social sign-up, choosing an identity provider from various sources (e.g., Google, Facebook).

Fourth, more weather providers will likely be available, or if the user simply wants to change the initial choice, the platform will give him the possibility to modify the selected providers or just change the color palette.

Fifth, the CIRC4FooD platform intends to add one more page called Sensor Dashboard, in which the user will have access to diagrams and a time series of data coming from all the sensors installed in the garden. The goal of this integration is for the user to have, in a more user-friendly manner, data from a more extended period of time. The user is going to have access to three types of time series:

- Daily
- Monthly

- Yearly

The aim of this extra feature is for the user to be fully informed and keep track of the measurements of each sensor. This way, the user will be able to collect all the agricultural tasks waiting to be executed and maximize the performance of the intelligent management system and the sustainability of urban farming.

V. DISCUSSION AND FUTURE WORK

A main challenge in smart farming platforms is consistency and compatibility among the utilized protocols, technologies, and actions. To address this issue, in this paper, we present the CIRC4FooD platform approach, which is an intelligent management system supporting sustainable urban agriculture. Existing smart farming platforms are not designed to support near-real-time data ingestion, quick analysis, and visualisation of large volumes of sensor data. For that purpose, one of the key components of the CIRC4FooD platform is its aptitude to deal with the high rate of sensor data by providing notification for fast and easily performed urban agriculture activities. The user interface of the platform allows complex data workflows to be collected, visualized, and executed without the need for programming skills.

Another important functionality of the platform is the ability to integrate automation on actuators based on the collected data from the sensors for optimizing food production in open or closed systems. The platform is designed in an adaptive way, able to fit in agricultural automation, enabling remote controlling and monitoring for irrigation scheduling to manage water usage for optimizing water resources. The automation functionality can be achieved through the connection with the developed dynamic rule engine, and its performance will be evaluated by monitoring the conditions of the sensors and actuators. Also, a quality assessment will be employed for adjusting the dedicated parameters of the dynamic rule engine with respect to the users' requirements.

The CIRC4FooD platform is created to enable the efficient management of urban vegetable gardens. In parallel, by engaging the user at different levels, it has the ambition to offer educational and awareness-raising advantages. In this paper, we present the ongoing work related to platform development describing the system and platform design.

An important feature requirement of the CIRC4FooD platform is to be able to scale, store, and visualize different kinds of sensors. The volume of data generated by the installed sensors is not a problem. Nonetheless, the velocity at which the data is produced is very high and results in a big set of sensor data and user data. Addressing this problem, we are aiming to do a performance analysis of the platform by measuring the load time from different sensors around the city of Trikala and specifying critical points, alongside with the adaptation of the platform from the users. The main objective of this performance analysis is to evaluate the scalability of the CIRC4FooD platform.

The CIRC4FooD platform will be tested in real-life settings in the coming months. The demonstration will last around six months in the city of Trikala in Greece and will start once the complete CIRC4FooD system is set up. Demonstrations will be performed in private gardens belonging to citizens of Trikala, in public, popular spaces of the city, and in several schools. The testing phase will be accompanied by a series of seminars of informative and educational character aimed at the users of the CIRC4FooD platform. It is expected that as a result of demonstration activities and received feedback, some features of the platform might be subject to change. Additionally, the user reward system will be implemented, and a repository for the users with information about how to use the platform and facts about the plants (e.g., preferences and common diseases) will be developed.

AKNOWLEDGMENT

This research is co-funded by Greece and the European Union through the Operational Programme Competitiveness, Entrepreneurship and Innovation 2014-2020 (EPAnEK) [code: T2EΔK-03899, project: CIRC4FooD – A circular economy inspired food production system].

REFERENCES

[1] D. Tsiakou, L. Łękawska-Andrinopoulou, M. Palazis-Aslanidis, V. Nousis, G. Tsimiklis, M. Krommyda, A. Amditis, E. Latsa, "CIRC4FooD Platform: An Intelligent Management System Supporting Sustainable Urban Agriculture", in Proceedings of The Tenth International Conference on Building and Exploring Web Based Environments (WEB 2022), IARIA Conference. [Online]. Available:http://thinkmind.org/index.php?view=article&articleid=web_2022_1_10_40005

[2] F. Stoessel, R. Juraske, S. Pfister, and S. Hellweg, "Life Cycle Inventory and Carbon Water FoodPrint of Fruits and Vegetables Application to a Swiss Retailer" Environ. Sci. Technol vol 46, 6, pp. 3253–3262, February 2012, doi: 10.1021/es2030577.

[3] FAO, 2021, "The State of the World's Land and Water Resources for Food and Agriculture – Systems at breaking point", Synthesis report 2021, Rome, https://doi.org/10.4060/cb7654en.

[4] United Nations, Department of Economic and Social Affairs, Population Division (2017), "World Population Prospects: The 2017 Revision, Key Findings and Advance Tables", Working Paper No. ESA/P/WP/248.

[5] V. Egli, M. Oliver, and E. Tautolo, "The Development of a Model of Community Garden Benefits to Wellbeing" Preventive Medicine Reports, Vol. 3, 2016, pp. 348-352, doi:10.1016/j.pmedr.2016.04.005.

[6] M. I. Cabral, S. Costa, U. Weiland, and A. Bonn, "Urban gardens as multifunctional nature-based solutions for societal goals in a changing climate." in: Nature-based solutions to climate change adaptation in urban areas. Theory and practice of urban sustainability transitions. N. Kabisch, H. Korn, J. Stadler, A. Bonn, Eds. Springer, Cham, pp. 237-253, 2017.

[7] M. Clarke, M. Davidson, M. Egerer, E. Anderson, and N. T. Fouch, "The Underutilized Role of Community Gardens in Improving Cities' Adaptation to Climate Change: A Review", People, Place and Policy 12 (3), pp. 241-251, Feb. 2019 doi: 10.3351/ppp.2019.3396732665.

[8] D. Pivoto et al.,"Scientific development of smart farming technologies and their application in Brazil", Information

Processing in Agriculture, Vol 5 (1), pp. 21-32, March 2018, doi: 10.1016/j.inpa.2017.12.002.

[9] N. Islam, M. M. Rashid, F. Pasandideh, B. Ray, S. Moore, R. Kadel, “A Review of Applications and Communication Technologies for Internet of Things (IoT) and Unmanned Aerial Vehicle (UAV) Based Sustainable Smart Farming”, Sustainability, 2021, https://doi.org/10.3390/su13041821.

[10] W. M. Júnior, T. T. B. Valeriano, and R. G. de Souza, “EVAPO: A smartphone application to estimate potential evapotranspiration using cloud gridded meteorological data from NASA-POWER system”, Computers and Electronics in Agriculture, Volume 156, 2019, pp. 187-192, https://doi.org/10.1016/j.compag.2018.10.032.

[11] A. J. Johnson et al., “Flavor-Cyber-Agriculture: Optimization of plant metabolites in an open-source control environment through surrogate modeling”, PLoS ONE 14(4), 2018. https://doi.org/10.1371/journal.pone.0213918.

[12] L. Li, X. Li, C. Chong, C. Wang, and X. Wang, “A decision support framework for the design and operation of sustainable urban farming systems”, Journal of Cleaner Production, Volume 268, 2020, 121928, https://doi.org/10.1016/j.jclepro.2020.121928.

[13] P. P. Jayaraman, A. Yavari, D. Georgakopoulos, A. Morshed, A. Zaslavsky, “Internet of things platform for smart farming: Experiences and lessons learnt”, Sensors 2016, https://doi.org/10.3390/s16111884.

[14] E. M. Ouafiq, A. Elrharras, A. Mehdary, A. Chehri, R. Saadane, M. Wahbi, “IoT in smart farming analytics, big data based architecture. ”, In Human Centred Intelligent Systems, Springer: Berlin/Heidelberg, Germany, 2021, pp. 269–279.

[15] M. A. Zamora-Izquierdo, J. Santa, J.A. Martínez, V. Martínez, A. F. Skarmeta, “Smart farming IoT platform based on edge and cloud computing”, Biosystems Engineering, Volume 177, 2019, pp. 4-17, ISSN 1537-5110, https://doi.org/10.1016/j.biosystemseng.2018.10.014.

[16] G. E. Mushi, G. D. M. Serugendo, P. Y. Burgi, “Digital Technology and Services for Sustainable Agriculture in Tanzania: A Literature Review”, Sustainability 2022, 14(4), 2415, https://doi.org/10.3390/su14042415.

[17] T.Howard, “Journey mapping: a brief overview.”, Communication Design Quarterly vol 2, (3), pp. 10–13, May 2014, doi:10.1145/2644448.2644451.

[18] Product School Inc., 2022 https://productschool.com/blog/product-management-2/experience/user-flows-vs-user-journeys/ (last visited: 15/02/2022).

[19] H. L. Dreyfus, S. E. Dreyfus, and T. Athanasiou. (1986), “Mind over machine: The power of human intuition and expertise in the era of the computer”, New York: Free Press.

[20] A. K. Sharma, R. Jain, D. Kumar, A. Teckchandani, V. Jain, “Implementation of reward-based methodology in web blogging environment”, Global Transitions Proceedings, Volume 2, Issue 2, 2021, pp. 579-583, https://doi.org/10.1016/j.gltp.2021.08.031 .

[21] MongoDB, Inc. 2021 https://www.mongodb.com/why-use-mongodb (last visited: 20/02/2022).

[22] OpenJS Foundation, Node.js https://nodejs.org/en/ (last visited: 20/02/2022).

[23] B. Paulin and P. O’Malley “Compost production and use in horticulture”, Department of Primary Industries and Regional Development, Western Australia, Perth. Bulletin 4746, 2008. Compost production and use in horticulture (agric.wa.gov.au).

[24] E. Vandaele, Vlaco, “Hygienisation requirements for composting”, Workpackage 5, Repost 2, Oct. 2019, available at: https://northsearegion.eu/media/16203/hygienisation_-for-soilcomher.pdf (last visited: 23/02/2022).

New Software Architecture for Monitoring Mobile Applications

Mobile Data Collection and Indexing for Mobile Application Monitoring

Mourlin Fabrice
Algorithmic, Complexity and Logic Laboratory
UPEC University
Cretéil, France
e-mail: fabrice.mourlin@u-pec.fr

Djiken Guy Lahlou
Applied Computer Science Laboratory
Douala University
Douala, Cameroon
e-mail: gdjiken@fs-univ-douala.cm

Laurent Nel
Leuville Objects
Paris-Saclay University
Paris, France
e-mail: laurent.nel@universite-paris-saclay.fr

Abstract—**The monitoring activity remains an activity that disturbs the system under control. We all try to minimize these disturbances in order to observe a behavior as close as possible to reality. In IT, this requires the implementation of a specific software architecture. Our use case concerns the monitoring of embedded applications on mobile devices for which the collected data sometimes contain errors that we want to explain. To this end, we seek to trace the important events of our calculations in order to qualify the anomalies in our processing. We have implemented a monitoring layer within mobile applications in order to perform intelligent monitoring on a set of mobile devices. We have defined a Big Data workflow to collect, index and store log data for submission to an artificial intelligence (AI) model. A crucial aspect of this collection relies on the use of partitioned topics and thus a better distribution of the data. With the increase of the data flow to be processed, the performance remains insufficient and we have opted for a persistence layer adapted to our data processing. We detect behavioral anomalies through the analysis of software logs deployed on embedded devices. Based on the patterns recognized in the logs, our AI model provides us with a sequence of system operations. These operations are then scheduled to redeploy a service, change a driver, perform a library update, etc. In the end, we build management reports every week for the maintenance team. These documents help track maintenance activities. They provide a record of important events such as equipment downtime or removal of obsolete services.**

Keywords— Big Data; indexing; log analysis; distributed application; AI model; storage efficiency; anomaly detection, explanatory report.

I. Introduction

Software monitoring remains without a doubt an active area of research. The diversity of situations is great because software supports very different deployment constraints. From the application installed on an application server, to the component running on a micro controller, the situations are very different. There is no single platform to monitor them. Therefore, everyone tries to specify his or her typical use case and cover a relevant sub-domain. The monitoring of mobile applications remains complex because it relies on operating systems with specific management rules. A crucial principle is to consider software administration as a facet of any mobile application. Thus, any installed application exposes resources that are used to evaluate its state. This one can be judged abnormal or not and actions are then implemented.

In order to have this information, developers use logging application programming interfaces (APIs) to transmit all the behavioral data. To make the information usable, the log messages usually have a format that facilitates information extraction. Each message usually has a priority level or severity hierarchy and a timestamp associated with an origin. The application administrator manages the level of expressiveness per module in order not to suffer from an excess of information.

The applications of log messages are numerous; they communicate an internal state to the users, but they can offer more like the reconstruction of a state. Database servers like Postgresql have log files containing the history of activity, from database creation to application connection triggers. Embedded applications have the same need. An Android application has access to a logging API, whether it is written in Java, Kotlin or C++. An Android logger corresponds to a variable in memory. It can be stored in a file or sent to a socket.

Log messages play a key role in the life cycle of a project. From the unit, integration, system and functional testing phases, logs are used to highlight processing steps. During development, they provide a view of the application's progress in terms of network, security, activity, etc. In the case of an embedded application, this data cannot be displayed because the device does not necessarily have a screen and it is useful to redirect the information into a persistence unit (memory card, memory stick, etc.). During the debugging phase, these same messages have a tag that allows them to be filtered, or even to specify a different level of severity from one packet to another to configure the feedback. In this study, the effort is focused on the analysis of the log messages used. For this purpose, we use topics on which each device publishes its messages. On the one hand, this allows partitioning the messages into categories; on the other hand, the processing of the published messages is easily distributed [1].

The difficulties linked to the analysis of logs are firstly linked to the volume of messages received. Indeed, this volume grows rapidly with the number of sources. Thus, in the case of monitoring embedded applications, when the number of devices increases, it is no longer possible in human terms to analyze the logs with serenity. The automation of this process is necessary.

A second difficulty is the flow of these messages, which depends on the use of the monitored applications. When the number of users increases, it is then necessary to set up a sampling of information. The third difficulty is that it is not uncommon for each embedded application to have its own log format, even if it is generated by the same API. In this case, it is necessary to consider a standardization of the formats during the collection in order to be able to extract information from different sources and to put them in a causal relationship. After processing, these log messages must also be stored in a persistence system capable of supporting variations in volume, velocity and variable format (3V of Big Data). This persistence system then becomes the underlying information center of the indexing module.

The set of properties related to the processing of log messages naturally leads us to consider this work as a Big Data workflow applied on a temporal scale. Indeed, it is essential to react to detected problems before they get worse. In this paper, we present the results of our work, which started more than two years ago with the Big Data prototyping of an anomaly prediction solution for Android applications. These mobile applications are used to take pictures of biology experiments and visualize them. Users can annotate each photo and reorganize the photos into a document related to a lab experiment. Users export their final document from the mobile device to a web server accessible from the Wi-Fi network at the experiment site.

The rest of this paper is structured as follows. Section II describes the works close to our domain. Section III provides a precise description of our use case. Section IV addresses the software architecture of our distributed platform and more precisely the partitioning of log data. Section V goes into finer details on our streaming approach, which includes an indexing step and a new storage layer oriented distributed document. Section VI focuses on our results, the impact on the maintenance task and the generation of explanatory report. The acknowledgement and conclusions close the article.

II. Related Works

Predictive log analysis is a widely studied topic. Part of the work focuses on enhancing the information itself. A second part concerns the use of this data to react, alert, and more generally automates a process.

A. Log analysis methods

Adam Oliner et al. describe, through several use cases, the information useful to report during execution for software monitoring [2]. They stress, among other things, the importance of adopting a consistent format throughout an application. They make the analogy between the follow-up of manufacturing on one meeting on a production line and the follow-up of the software activity, which is the subject of this work.

T. Yen et al. describe how to leverage distributed application logs for the detection of suspicious activity on corporate networks [3]. Their work highlights the use of the beehive tool for extracting information and producing easily exploitable messages. Analysis against a signature database is then possible.

S. He et al. present six methods for log analysis of distributed systems: three of them are supervised and three others are unsupervised. The authors make a comparative evaluation of these methods on a significant volume of log messages. They emphasize the strengths of software monitoring task automation [4] but the authors do not take into account the model storage regarding the properties of their data.

In more constrained fields such as real time, log analysis systems must be able to detect an anomaly in a limited time. B. Debnath et al. present LogLens that automates the process of anomaly detection from logs with minimal target system knowledge [5]. LogLens presents an extensible index process based on new metrics (term frequency and boost factors). The use of temporal constraint also intervenes in the recognition of behavioral pattern. Therefore, abnormal events are defined as visible in a time window while other events are not. This allows semi-automatic real-time device monitoring.

B. Log analysis and machine learning

The development of machine learning has greatly impacted the use of logs. Depending on the work, studies lead to the detection of anomalies or the discovery of software attacks.

Q. Cao et al. presented a work on web server log analysis for intrusion detection and server load reduction. The use of two-level AI model allowed them to increase the efficiency of their detection system. In this approach, the use of decision trees structures the log data [6].

W. Li considers that logs are a complement to the software-testing phase. Since the time allocated to testing is insufficient, he presents a failure diagnosis strategy based on the use of an AI model [7]. He provides a comparative study between several models.

There is a large body of work on network log analysis for various protocols including HTTP [8] or data-centric protocols such as Named Domain Networking (NDN) [9]. In all cases, the strategy is based on formatted messages where part of the information is filtered and then submitted to a model for prediction. Once again, the nature of the persistence system is not highlighted because the constraints of volume and data rate are not important.

C. Text indexing and storage

Textual data is widely used in Big Data, especially in linguistic analysis. It is mostly unstructured data, not referenced in a database. These data are therefore not interpretable by machines. S. Melink and S. Raghavan have built a distributed full text-indexing algorithm. They propose a storage scheme using an embedded database system of the H2 type [10]. Their results are promising on data from web browsing.

S. Melink and S. Raghavan have defined a novel pipelining technique for structuring the core index-building system. It substantially reduces the index construction time but the data and the index are stored in the same persistence layer. They provide a storage scheme for creating and managing inverted files using an embedded database system [11]. They present performance results obtained during experiments on a distributed web indexing testbed where we see that the data structure has no impact on the type of database used.

M. A. Qader and S. Cheng gather a very interesting comparative study of indexing techniques in the world of NoSQL databases. They allow a fast writing flow and fast searches on the primary key. Some of these persistence systems have added support for secondary indexes. These new indexes are useful for queries on non-primary attributes. Each NoSQL database usually supports a secondary index type. Their conclusion shows that there is no single system, which supports all secondary index types [12]. The authors highlight two classes of indexes: embedded indexes that belong to the storage system and autonomous indexes that are data structures distinct from the stored data. Their results show that none of these indexing techniques dominates the others. On the other hand, embedded indexes provide higher write throughput and are more memory efficient, while standalone secondary indexes provide faster response times when querying. In the end, the optimal choice of secondary index depends on the workload of the application, which is the case when analyzing log messages.

D. Reporting of artificial intelligence prediction model

In order to obtain a set of guidelines for the use of predictive machine learning models, it is essential to build regular reports on the quality of predictions. In the context of clinical experiments, W. Luo et al. published a rulebook for AI model development [13].

P. Henderson et al. present a systematic reporting of the energy and carbon footprints of machine learning. The authors' goal is to adapt an efficient reinforcement learning strategy and explain the reinforcement learning events [14]. Events from the environment are associated with their evaluation and recorded. The report traces the life cycle of the AI model.

L.M. Stevens et al. present a recommendation for transparent and structured reporting of Machine Learning (ML) analysis results specifically directed at clinical researchers [15]. Their goal is to convince many clinicians and researchers who are not yet familiar with evaluating and interpreting ML analyses. The model provides evaluation measures that offer a means of comparison between models and underlying strategies.

D.P. Dos Santos et al. take a similar approach to the analysis of radiological images. Their quality is uneven and it becomes difficult to provide a reproducible analysis approach. It then becomes essential to build reports to explain the state of the AI model that led to certain predictions. The authors explain how to structure to help build a post analysis explanation [16].

The use of AI models relies on data from persistence systems. The use of Big Data processing aims to bring data from a data lake into a data lab. This data lab usually consists of a NoSQL database and events such as insertion and update have a strategic role on the life cycle of the associated model. S. Afonin, et al. describe an automatic report generation system based on the database activity. They use a zero-code solution where the underlying software is either Jasper Report [17]. The interest is the provision of a report in a format adapted to the use (Web, pdf, etc.).

III. Use Case Description

A. Context Description

In biology trainings, many experiments are done where students are asked to prepare, perform and follow up. In this context, mobile devices are provided to take pictures, record sounds, or even use the device's sensors to collect data. To save different documents in the memory of the mobile device, a software suite is installed. It allows the authentication of the user, the dating of each collected information and the transfer at the end of the experiment to a server for validation.

During an experiment, all the devices are connected to the laboratory Wi-Fi network. This connection authorizes data exchange with the laboratory server, which will receive all the students' data at the end of the experiment for validation by the supervisor. This network connection is also used to send log messages to monitor the activity of each mobile device. This concerns the capture of information: taking a picture or recording a single comment or a short video. This type of recording is not often used during an experiment because several students are monitoring the same experiment and this leads to noise pollution for the other participants.

The analysis of an experiment by a group of twenty students takes place over a period of one-day maximum in the same experimentation room. This means that the connection is made with the same access point for all devices. Even if the batteries are initially charged, it is possible at any time to have a recharging point in the room.

Laboratory observation sessions can be short in the early grades, such as showing the release of gas bubbles by an aquatic plant. Students construct a document to highlight the conditions of this phenomenon and then make a video to support their comments. Then, they observe the role of light and measure its value with the light sensor of the Android tablet. A second video will show the release of gas bubbles by an aquatic plant in the presence of light. In the absence of light, the students make a comparison with pictures.

In the lab room, a group of students follows an experiment with one tablet per student. Each tablet allows a student to take pictures or videos in order to build his observations of an experiment. The student first saves them locally on the tablet. A typical scenario consists of one Wi-Fi access point per room, a set of mobile devices and a remote storage server for document backups at the end of the session. This scenario is to be multiplied by the number of groups, possibly in parallel in different lab rooms. Two properties are thus highlighted: on the one hand, a local authentication phase on the mobile device, on the other hand a centralized storage server (see Figure 1). In addition, the lab room has a laptop for the teacher and a shared printer. The teacher thus has access to the documents that have been saved on the storage server. Furthermore, he can observe the tablets connected to the lab network and record the addresses of the tablets participating in the study.

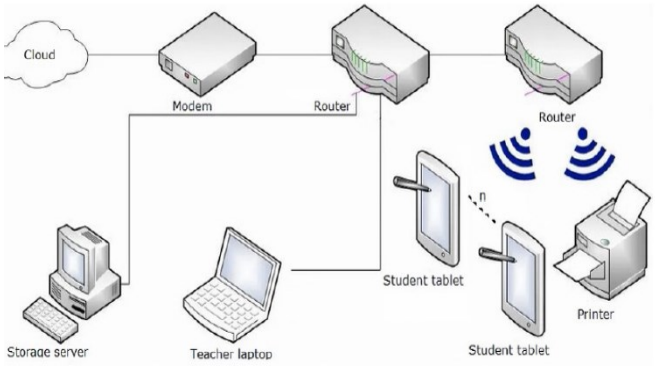


Figure 1. Network diagram of a laboratory room.

The first router provides a Wi-Fi network to the devices in the lab room. The second router provides access to the Big Data workflow, which starts with a set of message queues available to mobile devices. Without this bridge between the lab room and the data center, we would not have a mobile data responsive architecture.

B. Scenario description

In order to describe a nominal scenario more precisely, let us take the case of a student from the beginning of a lab session to the submission of a document at the end of the session.

Figure 2 describes the general flow of the scenario in the Unified Modeling Language (UML) notation; it concerns an observation session (Lab Session). The biology teacher manages this session. Each student manages their own documents locally on their local device. Thus, the student takes notes, photos, videos and measurements via the available sensors. For example, infrared radiation allows the detection of the closest objects. This provides appropriate distance measurements during experiments. When his work is finished or the teacher has closed the session, a student prepares his final document, signs it and deposits it on the storage server.

The storage server receives the work of students for each experiment. The teachers will consult the works to make their post experimentation evaluation.

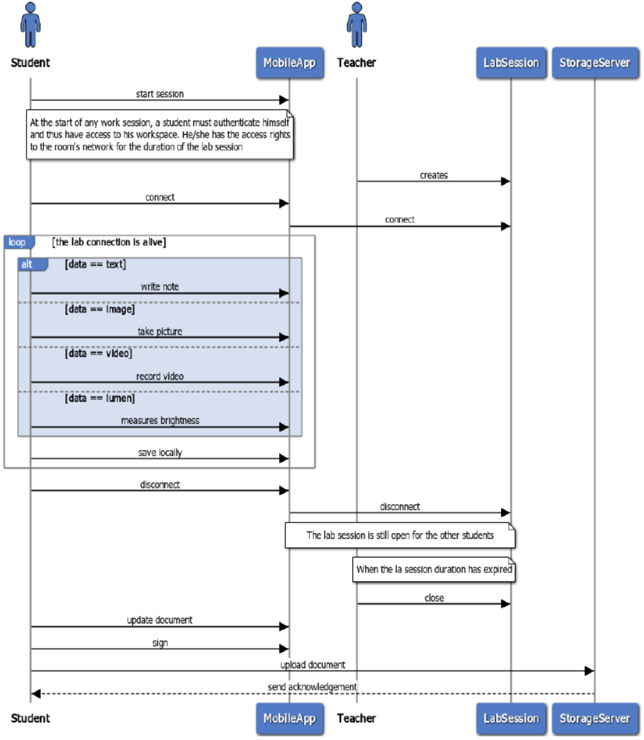


Figure 2. Nominal scenario during an experiment in a lab room.

We do not address in this work the management of student-provided materials throughout the academic year. We focus on the aspects related to an experimentation and the follow-up of this activity by processing the associated log messages.

IV. Software Architecture

If the software architecture of the business part is very simple, it is only the entry point of the information collection, which gathers the log data on the storage server. The analysis of these logs is more complex because it takes into account additional constraints: the arrival of log data continuously, the need to impose a data schema to index the information, refine the search for information and the detection of anomalies.

A. Client application

In order to collect information from the activity of the actors in the scenario in Figure 2, the log system of the devices is enabled by the students and the teacher. Our goal is to collect and cross-reference information from the various sources. Thus, it is essential to monitor the events related to the management of the laboratory sessions. In addition, any event related to an information capture or modification is useful.

1) Mobile application

The *MobileApp* instance in Figure 2 is an Android component installed on each tablet. The set of class is written in Java using the log API specific to this system. In the business part, we have defined a message format in order to easily extract the information. The creation of the log messages occurs by using the *android.util.Log* class, which allows not only to prefix with a semantic tab, but also to add a severity level. Thus, from a set of messages, a regular expression filters the relevant results to focus on the essential data.

In addition to the business events, in this effort, we have traced the memory events provided by the garbage collector, the transmissions and receptions of information from the http network. Moreover, we used the Android Mobility Management API to define usage profiles such as Student profile. It allows business apps and data to be stored in a separate, self-contained space within a device. The teacher has full management control over the applications, data, and Student profile settings on the device, but has no visibility or access to the device's personal profile. This strong separation allows teachers to control *MobileApp* data and security without compromising student privacy if they are using applications other than those intended for the biology course.

We have developed a Device Policy Controller (DPC), which logs network activity. Network activity logging helps us detect and track the spread of malware on tablets. Our DPC uses network logging APIs to report the Transmission Control Protocol (TCP) connections and Domain Name System (DNS) lookups from system network calls.

To further process the logs on our Big Data cluster, we have configured DNS deny lists to detect and alert for suspicious behavior. We have enabled Android network logging to record every event from the *MobileApp* application. It uses the system's network libraries. Network logging records two types of events: DNS lookups and network connections. The logs capture every DNS query that resolves a host name to an IP address. Other supporting DNS queries, such as name server discovery are not logged. The Network Activity Logging API presents each DNS lookup as a DnsEvent instance. Network logging also records an event for each connection attempt that is part of a system network request. The logs capture successful and failed TCP connections, but User Datagram Protocol (UDP) transfers are not recorded. The Network Activity Logging API presents each connection as a ConnectEvent instance. This entire network log configuration is complex, but grouped in a specific concrete class named DevicePolicyManager. The configuration is taken into account asynchronously and it is important to validate it before distributing the tablets to students at each software update.

2) Mobile component

The component deployed on the teacher's laptop is a traditional Java component (version 11) also configured with a message format and a log level. This provides a trace of important events that take place on this workstation. Log analysis is the fastest way to detect what went wrong, which makes logging in Java essential to ensure the performance and health of our distributed application. The goal is to minimize and reduce any downtime, to reduce the mean time to repair.

We used the slf4j library because it represents a simple and highly configurable API. In particular, we have configured the directory where the log messages are saved as well as the expression to generate the file names with the date. The size of the messages is voluntarily limited, so that the subsequent collection is always done within a reasonable time. In addition, the stack trace is provided for any anomaly. Finally, the structure of all logging events follows a pattern consistent with the *MobileApp* component. We have added a log converter to hide some information such as student IDs. It is important that sensitive information is not traced because this data is then transmitted to our Big Data cluster for analysis.

B. Server application

The server application part is deployed on the storage server. This component, also written in Java (version 11), contains the implementation of Web service allowing on the one hand to receive the documents of the students but also to acknowledge the receptions. This part is developed with the Spring Boot library. We use intensively the Spring configuration for the logs, it is indeed a simple way to define a different log level from one package to another but also for the persistence aspects. The database is Postgresql version 10. This database server is used for the persistence of data resulting from the work of students and teachers. As in the previous section, the location of the log files for our server component or for the Postgresql server is imposed. As an example, we record the trace of any http request received by our Web services. The headers are kept as well as the response headers. The version of the http protocol used is http/1.1. In the same way as for the Laptop component, we have imposed a log message format.

C. Big Data architecture

This section focuses on describing our Big Data workflow from collection to building our AI model. We wanted to automate our approach as much as possible because any human intervention leads to blockages or even loss of information.

In this section, we have made technical choices leading to the use of open source software. First, we use the Apache Kafka message queue server. Its role is to receive log messages coming from mobile devices by classifying them by topic. These topics are distributed and Apache Kafka acts as a mediator between two worlds: the mobile device and the Big Data workflow. Secondly, we use the Apache Flume server, which allows defining routes between two or more software. Thus, data can flow from software A to software B. In some aspects, it plays the role of an ETL (Extract, Transform, Load). Third, we use Apache Spark to develop and run our Big Data programs. This library helps build in-memory SQL tables that are then stored in a database such as MongoDB which is a document-oriented NoSQL database. Its strong point is to allow the use of simple foreign keys. Finally, Apache Spark contains a sequencer that manages the execution of the built programs. It plays the role of a Big Data engine. The fourth tool is the Apache SolR server, which plays the role of index creator of the data from the previous SQL tables. This means that we define our data index calculation algorithm with the SolR library. At runtime, Apache SolR only stores the indexes while the data is stored in the MongoDB database. The searches are thus faster. The fifth and last tool is Jasper Report from Tibco. It is a tool for building and automatically generating reports. It plays the role of an information distributor for end users.

1) Data collection

This part deals with the collection of log files in order to send them to a Kafka queue. These Kafka files are the entry points of our Big Data cluster. Because there are 3 different types of components, our best choice was to build an event-based collection based on scenario actions. For the *MobileApp* part, the logs are recorded locally on the device. The sending of the information to a Kafka topic is done when the student sends his final document to the storage server. This approach reduces the number of accesses to the Kafka topic server. Thus, the access point of the lab room has been used to send an http request with an attachment part (the document). This sending is also present in the logs so that the next time only a request is sent, not the same data but only the new ones.

The same approach is used for the laptop component. We use an event-based approach. When the lab session is closed, the logs on the laptop are sent to a Kafka file of the same topic. The message volume is lower, but the information is essential when associating with the logs of the mobile devices.

For the storage server, a repetitive task was our best bid because this central point does not reveal any particular interaction but a continuous flow of data. A cron table was used to collect logs from the Postgresql server and the server component to a Kafka file of the concerned topic. Data are automatically routed to the Kafka server where the topics are managed. All log message traffic can be observed via dedicated Kafka system scripts or by using existing JMX components (Java Message Extension).

2) Big Data analysis

A Big Data cluster that is built from Hortonworks Data Platform (HDP) 3.0.1 virtual machines is used to perform log analysis. This solution offers the advantage of deploying software from the Hadoop ecosystem while remaining open to other installations. Moreover, the Ambari console allows a simple configuration of servers such as Kafka for topics or Flume for routes. Our software architecture for Big Data is based on two software routes from Kafka topics to the persistence system. In a first version presented at the AllData 2022 conference, our persistence layer for log messages was based on the HBase server. Indeed, it is installed by default in the HDP virtual machine and offers a data mapping on top of HDFS Hadoop Distributed File System). This type of column family oriented database has natural advantages for data parallelization. HBase is designed to work with key/value data and random read and write access patterns. Its Java library is easy to use but the types of data accepted are poor or very poor. It is penalizing because our data, essentially textual, also have integer and real fields following our analysis and indexing. The consequences are a cost in time increasing with the volume of processed data. While the times were acceptable for the first full-scale tests, when the number of course sessions increased, we observed very long processing times. Thanks to the monitoring of the VMs, we were able to understand the origin of the waits: access to the region servers, encoding and decoding phases of large numbers of data, overly complex queries, etc.

Figure 3 shows the layer components of our project. Deployed on the lab room platforms, we developed Kafka producers for the peripherals, the teaching laptop and the storage server. All these producers issue log messages in a Kafka topic that is partitioned on the server. This improves the access time to the information.

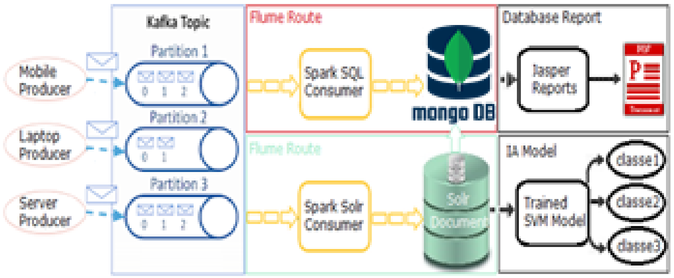


Figure 3. Big Data Software architecture.

The topic partition is the unit of parallelism in Kafka. On both the producer and the broker side, writes to different partitions are done fully in parallel. At the output of the Kafka topics, two Flume routes have been defined within this experiment, each managed by an agent. A first route (red on Figure 3) consumes the messages in order to transform them for some residual format differences and store them in a document-oriented database, MongoDB, installed on the cluster. A second route (green on Figure 3) consumes the messages to index them according to a Solr data schema. Each persistence system has its own role: MongoDB keeps the structured log data and Solr keeps the indexes on these data to enrich the searches. We consider MongoDB and Solr as two data sources accessible from Spark components. The Spark SQL API is easily used to write to MongoDB collections on a Hadoop cluster. In contrast, our Spark to Solr consumer does not have such an easily accessible API and we used Solr Cloud REST services for our updates.

The data indexed by Solr enables our system to classify the messages in order to carry out maintenance operations on the various materials. A relevant option here was a linear classifier with margin calculation. In fact, in several evaluations of AI models, it is established that in the category of linear classifiers, the Support Vector Machine (SVM) are those that obtain the best results. Another advantage of SVMs, and one that is important to note, is that they are very efficient when there is little training data: while other algorithms would fail to generalize correctly, SVMs are observed to be much more efficient. However, when there is too much data, the SVM tends to decrease in performance.

In order to understand the MongoDB events and their distribution on the cluster, we have defined a report template to generate a pdf report. It summarizes the activities by collection, their events, in particular the use of shards. The use of a template guarantees the scalability of these reports according to the evolutions of the consumer SQL Spark. We added a page header with a table name and the current edition date and a page footer with the page number. The column header band is printed at the beginning of each detail column with the column names in a tabular report. This means the part name of a log message.

3) Log Data storage

A first Spark consumer (named "Spark SQL consumer") has an essential task to recognize and process the contents of the file and load them into an SQL table in memory, perform filter operations and put them in a common format. Then, the route continues with a backup of these data in MongoDB collections. The role of this Flume route is to store structured information in a document-oriented database (the red route in Figure 3). In this effort, we experimented keeping software routes with Flume for event routing and defined Kafka topics to ensure decorrelation between components. This makes it possible to simplify the management of components, among other things for software updates. In addition, the Kafka API allows more controls on the management of messages associated with a topic; for example time management. We have added rules to ensure that a received message is processed within an hour (from a configuration file). In that case, the system raises an alert and the data saved in the local file system.

A Flume agent is an independent daemon process, which manages the red route. The Flume agent ingests the streaming data from the Kafka topic source to the Spark SQL sink. The channel between the source and the sink is a temporary storage. It receives the events from the Kafka source and buffers them until they are consumed by Spark sinks. It acts as a bridge between the source and the sinks. We have added a Flume interceptor to decide what sort of data should pass through to the channel. It plays first a filter role in case of unsuitable data from the Kafka source and inserts the time in nanosecond into the event header. If the event already contains a timestamp, it will be overwritten with the current time.

In a previous prototype of this project, we noticed the performance limits of a storage solution based on HBase. Although it is distributed on the cluster, the distribution of data in column families was not well adapted to our data types and our queries when searching for information for the AI model construction. HBase remains ideal for very large-scale use cases but with a simple format. This is not the case with our formatted log messages. HBase offers very fast searches if we are looking for information on a particular key, but MongoDB provides a much richer model that allows us to follow the evolution of information during its life cycle. MongoDB's data model is relational and allows us multi-document ACID transactions, and its query language is rich.

The HBase persistence system was uninstalled from the reference VM to install the MongoDB suite with monitoring and query software. Therefore, MongoDB becomes a part of the Ambari stack. We monitor and manage this service remotely via REST API. MongoDB offers a wide choice of cluster types to create a specific cluster. Each type represents feature limitations and space limitations of the specific cluster. We chose a 10 GB space for our new prototype. Then, we created accounts for the projects that use the collections in the database, with an associated authentication method. We have customized the access privileges to the database. MongoDB schema design works a lot differently than relational schema design because there is no rule and no obvious process. Two different applications that use the same exact data have different schemas if the applications are used distinct manners.

We have created our collections by using JSON schema (JavaScript Object Notation). If this definition comes from the structure of the log messages, we have added additional fields to track the processing (date of analysis, date of inclusion in model, etc.). We have favoured the integration of data in the same collection to reduce joins. We only use tables of a size fixed by the schema [Table I]. Finally, our data schema depends essentially on the access to our data during the construction of the AI model. We used references to data only to avoid duplication of data and avoid data cycles. Log entries are written as a sequence of key-value pairs, where each key indicates a log message field type, such as “origin” or "severity", and each corresponding value records the associated logging information for that field type.

TABLE I. STRUCTURE OF THE MAIN COLLECTION

Field name	Type	Description
ts	DateTime	Timestamp of the log message in ISO-8601 format.
severity	String	Short severity code of the log message.
id	Integer	Unique identifier for the log statement.
context	String	Name of the thread that caused the log statement.
message	String	Log output message passed from the server or driver.
attributes	Object	Optional: one or more key-value pairs for additional log attributes (limit to 10)
tags	Array	Strings representing any tags applicable to the log statement (limit to 10 strings)
origin	String	Network description of the message source.
creation	DateTime	Timestamp of the log message insertion event in MogoDB
consumption	DateTime	Timestamp of the consumption event of the log message into the AI model
report	DateTime	Timestamp of the consumption event of the log message into the generated report

Our Spark SQL consumer uses the Spark SQL module to store data in a MongoDB database whose schema is structured in collections of documents. The labels of these key/value pair are involved in the data schema of the second Spark consumer. Our MongoDB cluster is used in data replication mode. This means that replication is done on a group of MongoDB servers that hold copies of our data. This is a critical property for deployment as it ensures high availability and redundancy, in case of outages and maintenance periods.

4) Log Data indexing

In parallel, another route has the role of indexing the data from the logs (green route in Figure 3). From the same Kafka source, a second Spark consumer (named "Spark Solr consumer") takes care of data indexing while respecting the Solr schema. The index is updated for the query steps and then the use of a model for the prediction of maintenance tasks. Solr Cloud is the indexing and search engine. It is completely open and allows us to personalize text analyses. It allows a close link with MongoDB, database so the schemas used by both tools are designed in a closely related way. On our Big Data cluster, the Solr installation is also distributed. In that context, we have four shards with a replication rate equals to three. This allows us to distribute operations by reducing blockages due to frequent indexing. We have configured, not only the schema, but also the data handlers (schema.xml and solrconfig.xml files).

Our schema defines the structure of the documents that are indexed into Solr. This means the set of fields that they contain, and define the datatype of those fields. It configures also how field types are processed during indexing and querying. This allows us to introduce our own parsing strategy via class programming. Having evolved our persistence layer in order to have a richer data model, it seems natural to choose a data schema compatible between MongoDB on the one hand and Apache Solr on the other hand. The data processing being separate, we had to adapt ourselves in order to make them match as well as possible. For example, the Datetime type of MongoDB plays the same role as the DatePointField type of Solr but its interpretation is not identical.

The Spark Solr consumer uses the Spring Data and SolrJ library to index the data read from the Kafka topic. It splits the data next to the Solr schema where the description of each type includes a "docValue" attribute, which is the link to the MongoDB identifier. For each Solr type, our configuration provides a given analyzer. We have developed some of the analyzers in order to keep richer data than simple raw data from log files. Finally, the semantic additions that we add in our analysis are essential for the evaluation of Solr query. Likewise, we store the calculated metrics in MongoDB main collection as an attribute for control. SolrCloud is deployed on the cluster through the same Zookeeper agents. Thus, the index persistence system is also replicated. We therefore separate the concepts of backup and search via two distinct components. This reduces the blockages related to frequent updates of our Mongo database [18].

At the beginning of our Solr design, we have built our schema based on our data types. Some of them were already defined, but some others are new. In addition, we have implemented new data classes for the new field types. For example, we used RankFieldType as a type of some fields in our schema. It allows us to manage enumeration values from the log message. Then, it becomes a sub class of FieldType in our Solr plugin. We have redesigned Solr filters so that they can be used in our previous setups. Our objective was to standardize the values present in the logs coming from different servers. Indeed, the messages provide information of the form <attribute, value> where the values certainly have units. However, the logs do not always provide the same units for the same attribute calculation. The analysis phase is the place to impose a measurement system in order to be able to compare the results later. The development pattern proposed by SolrJ is simple because it proposes abstract classes like TokenFilter and TokenFilterFactory then to build inherited classes. Then we have to build a plugin for Solr and drop it in the technical directory agreed in the installation of the tool [19].

5) Model factory

In Artificial Intelligence, Support Vector Machine (SVM) models are a set of supervised learning techniques designed to solve discrimination and regression problems. SVMs have been applied to a large number of fields (bioinformatics, information research, computer vision, finance, etc.) [20]. SVM models are classifiers, which are based on two key ideas, which allow to deal with nonlinear discrimination problems, and to reformulate the ranking problem as a quadratic optimization problem. In our project, SVMs can be used to decide to which class of problem a recognized sample belongs. The weight of these classes if linked to the Solr metrics on these names. This amounts to predicting the value of a variable, which corresponds to an anomaly.

All filtered log entries are potentially useful input data if it is possible that there are correlations between informational messages, warnings, and errors. Sometimes the correlation is strong and therefore critical to maximizing the learning rate. We have built a specific component based on Spark MLlib. It supports binary classification with linear SVM. Its linear SVMs algorithm outputs an SVM model [21]. We applied prior processing to the data from our Mongo collections before building our decision modelling. These processes are grouped together in a pipeline, which leads to the creation of the SVM model with the configuration of its hyper-parameters such as weightCol. Part of the configuration of these parameters comes from metrics calculated by our indexing engine (Figure 2). Once created and tested, the model goes into action to participate in the prediction of incidents. We use a new version of the SVM model builder based on distributed data augmented. This comes from an article written Nguyen, Le and Phung [22].

6) Report generation

Jasper Report library allows us to build weekly graphical reports on indexing activity. MongoDB events are collected for displaying. The goal is to correlate the volumes of data saved in the database with the updates of the AI model. We would like to refine this report template in order to have metrics to decide on the model update. Currently, only MongoDB movements are represented graphically. Based on an MongoDB handler, we handle the change events at runtime and send data beans to the Jasper Report Server.

Jasper Report has its own query language in JSON format. It allows you to specify the data to be extracted from MongoDB. The connector converts this request into appropriate API calls and uses the MongoDB Java connector to query our MongoDB cluster. In particular, it is possible to perform aggregation in the form of map-reduce, but more efficient than a simple pipeline. The map-reduce key specifies a map-reduce operation whose result will be used for the current query. The collection name specifies the target collection. This optimized extraction gives us better performance when building the AI model.

V. Data Streaming Part

A. Data streaming

Our component called Spark SQL Consumer contains a Kafka receiver class, which runs an executor as a long-running task. Each receiver is responsible for exactly one input discretized stream (called DStream). In the context of the first Flume route, this stream connects the Spark streaming to the external Kafka data source for reading input log data.

As an example in Table II, we provide an example of a log messages from a Kafka topic called “RedTopic”:

TABLE II. Message from Kafka topic

```
{
  "ts": "2020-10-02T18:10:22Z",
  "severity": "INFO",
  "id": 192674,
  "context": "kafka.server.KafkaServer",
  "message": "1000ms metadata for topic =
RedTopic   partition   =   part00002   not
propagated to all brokers",
  "attributes": {
        "process-id": 4122,
        "event-type": "ReadEvent"
  },
  "tags": [
       "Startup"
  ]
}
```

Structured Streaming, allows us to write streaming queries in the same way as batch queries. Spark streaming uses micro-batch processing, which means that data is delivered in batches to executors. If the executor idle time is less than the time required to process the batch, the executor is constantly added and then removed. If the executor idle time is longer than the batch time, the executor is never deleted. Therefore, we have disabled dynamic allocation by defining when running streaming applications. A Kafka partition is only be consumed by one executor, one executor consumes multiple Kafka partitions. This is consistent with Spark Streaming.

B. Filtered log strategy

Because the log data rate is high, our component reads from Kafka in parallel. Kafka stores the data logs in topics, with each topic consisting of a configurable number of partitions. The number of partitions of a topic is an important key for performance considerations as this number is an upper bound on the consumer parallelism. If a topic has N partitions, then our component can only consume this topic with a maximum of N threads in parallel. In our experiment, the Kafka partition number is set to four.

Since log data are collected from a variety of sources, data sets often use different naming conventions for similar informational elements. The Spark SQL Consumer component aims to apply name conventions and a common structure. The ability to correlate the data from different sources is a crucial aspect of log analysis. Using normalization to assign the same terminology to similar aspects can help reduce confusion and error during analysis [21]. This case occurs when log messages contain values with different units or distinct scales. The log files are grouped under topics. We apply transformations depending on the topic the data come from. The filtered logs are cleaned and reorganized and then are ready for an export into a MongoDB instance.

In the next step, the Spark SQL Consumer component inserts the cleaned log data into memory data frames backed to a schema. We have defined a mapping between MongoDB and Spark tables, called Table Catalog. There are two main difficulties of this catalog.

a) The identifier definition implies the creation of a specific index generator in our component.

b) The mapping between table column in Spark and the document in MongoDB needs a component for dynamic data frame creation with Spark SQL.

The MongoDB sink exploits the parallelism on the set of MongoDB nodes. A MongoDB replication is enforced by providing data redundancy and high availability over more than one MongoDB server. In addition to fault tolerance, replica sets also provide extra read operations capacity for the MongoDB cluster, directing clients to the additional servers, subsequently reducing the overall load of the cluster. A MongoDB cluster has a primary node and a set of secondary nodes in order to be considered a replica set. One primary node exists, and its role is to receive all write operations. All changes on the primary node's data sets are recorded in a special capped collection called the operation log (oplog). The role of the secondary nodes is to then replicate the primary node's operation log and make sure that the data sets reflect exactly the primary's data sets.

The driver Spark generates tasks per data set. The tasks are sent to the preferred executors collocated with a Mongo server, and are performed in parallel in the executors to achieve better data locality and concurrency. By the end of an exportation, a timed window a log data are stored into MongoDB collections.

C. *Index construction and query*

The strategy of the Spark Solr Consumer component deals with the ingestion of the log data into Apache Solr for search and query. The pipeline is built with Apache Spark and Apache Spark Solr connector. Spark framework is used for distributed in memory compute, transform and ingest to build the end-to-end pipeline.

The Apache MongoDB role is the log storage and the Apache Solr role is the log indexing. Both are configured in cloud mode Multiple Solr servers are easily scaled up by increasing server nodes. The Apache Solr collection, which plays the role of a SQL table, is configured with shards. The definition of shard is based on the number of partitions and the replicas rate for fault tolerance ability. The Spark executors run a task, which transforms and enriches each log message (format detection). Then, the Solr client takes the control and send a REST request to Solr Cloud Engine. Finally, depending on the Solr leader, a shard is updated.

We use also Solr Cloud as a data source Spark when we create our ML model. We send requests from Spark ML classes and read results from Solr (with the use of Solr Resilient Distributed Dataset (SolrRDD class). The pre statement of the requests is different from the analysis of the log document. Their configuration follows another analysis process. With Spark SQL, we expose the results as SQL tables in the Spark session. These data frames are the base of our ML model construction. The metrics called Term Factor (TF) and Inverse Document Frequency (IDF) are key features for the ML model. We have also used boost factor for customizing the weight of part of the log message. The data in the database is updated with the addition of attributes to the documents that have just been read. Finally, the use of this data for the AI model is recorded in order to distinguish it from new information to come.

VI. Results and Task Maintenance

We have several kinds of results. A part is about our architecture and the capacity to treat log messages over time. Another part is about the classification of log messages. The concepts behind SVM algorithm are relatively simple. The classifier separates data points using a hyperplane with the largest amount of margin. In our working context, the margin between log patterns is a suitable discriminant.

A. *Data features*

For our tests, we used previously saved log files from a month of application server and database server operations. We were interested in the performance of the two Spark consumers: For Spark SQL Consumer, the volume of data to analyze is 102.9 M rows in HBase. To exploit this data, we used a cluster of eight nodes on which we deployed Spark and MongoDB. The duration of the tests varies between 38 minutes and 3 hours and 51 minutes.

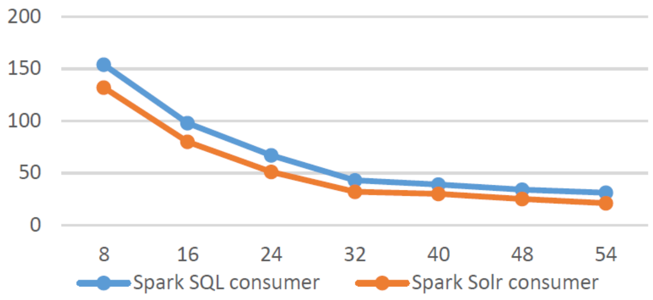


Figure 4. Spark consumer runtime versus number of partitions.

For Spark Solr Consumer, the volume of data indexed is 100.5M rows indexed in about an hour. The number of documents indexed per second is 35k. We only installed Solr on four nodes with four shards and a replication rate of three. We have seen improved results by increasing the number of Spark partitions (RangePartitioner). At runtime for our data set based on a unique log format, the cost of Spark SQL consumer decreases when the partitioning of the dataset increases, as illustrated in Figure 4. The X-axis represents the partition number and the Y-axis represents the time consuming. We have to oversize the partitions and the gains are much less interesting.

SVM offers very high accuracy compared to other classifiers such as logistic regression, and trees. There are several modes of assessment. The first is technical; it is obtained thanks to the framework used for the development (Spark MLlib). The second is more empirical because it relates to the use of this model and the anomaly detection rate on a known dataset. The analytical expression of the features precision, recall of retrieved log messages that are relevant to the find: Precision (1) is the fraction of retrieved log messages that are relevant to the find:

$$precision = \frac{|\{relevant\ \mathrm{log}\ messages\} \cap \{retrieved\ log\ messages\}|}{|\{retrieved\ \mathrm{log}\ messages\}|} \qquad (1)$$

Recall (2) is the fraction of log messages that are relevant to the query that are successfully retrieved:

$$recall = \frac{|\{relevant\ \mathrm{log}\ messages\} \cap \{retrieved\ log\ messages\}|}{|\{relevant\ \mathrm{log}\ messages\}|} \qquad (2)$$

$$F_\beta = (1+\beta^2) * \frac{precision*recall}{(\beta^2*precision)+recall} \tag{3}$$

In Table III, we have four classes and for each class we compute three metrics: true positive (tp), false positive (fp) and false negative (fn). For instance, for the third class, we note these numbers tp3, fp3 and fn3. From these values, we compute precision by label, recall by label and F-score by label.

TABLE III. SVM MODEL MEASURES

Class number	Metrics		
	Precision by label	*Recall by label*	*F1 score by label*
0.0000	0.8158	0.8901	0.8966
1.0000	0.9110	0.9810	0.9910
2.0000	0.8545	0.7857	0.8515
3.0000	0.8524	0.7589	0.8331

Our prediction models are similar to a multiclass classification. We have several possible anomaly classes or labels, and the concept of label-based metrics is useful in our case. Precision is the measure of accuracy on all labels. This is the number of times a class of anomaly has been correctly predicted (true positives) normalized by the number of data points. Label precision takes into account only one class and measures the number of times a specific label has been predicted correctly normalized by the number of times that label appears in the output. The last observations are:

- Weighted precision = 0.9017
- Weighted recall = 0.9318
- Weighted F1 score = 0.9817
- Weighted false positive rate = 0.04009

Our results for four classes are within acceptable ranges of values for the use of the model to be accepted.

The test empirical phase on the SVM model was not extensive enough to be conclusive. However, our results suggest that increasing the number of log patterns deteriorates the performance. In addition, we defined a finite set of log patterns for a targeted anomaly detection approach.

B. Reporting

1) From storage activities

MongoDB is not well designed for analytics purpose, we have set up a SQL data warehouse, and we have used Apache Camel to load data into a H2 warehouse. Apache Camel is a simple ELT (Extract Load and Transform) data from a data source into a SQL sink [23]. Thanks to the events performed on our collections, we can visualize the traffic variations on our collections according to the sampling duration we have chosen (Figure 5). We also monitor AI model updates following log message collections. These results are first displayed in the weekly report. We also count events related to the use of damaged replicas as well as all incidents during our data retrievals. In the end, we currently get a set of tables summarizing the activity on the data in our persistence layer. As a first observable result, we see that the number of unavailability is much lower than in our previous HBase-based prototype.

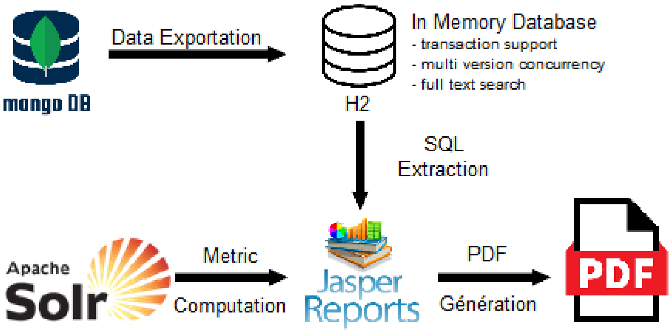


Figure 5. JSR and two data sources

2) From indexing activities

We have created a custom data source to connect to Apache Solr, therefore we are able to retrieve data and provide them back in following the JRDataSource interface of Jasper Report. With this access point, we have extracted metrics about the document cache and Query result cache. Both give an overview of the Solr activities and is meaningful for the analysts. We have deployed the CData JDBC Driver on Jasper Reports to provide MongoDB data access from reports. We have found that running the underlying query and getting the data to our report takes the most time. When we generate many pages per report, there is overhead to send that to the browser.

For the reporting phase, we have developed two report templates based on the use of a JDBC adapter. With system requests, we collect data about the last events (Create, Update, and Delete). From these H2 view, we have designed the report templates with cross tables. For the storage phase, we compute and display the number of Update events per timed window or grouped over a period. We periodically updated the data across report runs. We export the PDF files to the output repository where a web server manages them.

VII. CONCLUSION AND FUTURE WORK

We have presented our approach on log analysis and maintenance task prediction. We showed how an index engine is crucial for a suitable query engine. We have developed specific plugin for customizing the field types of our documents, but also for filtering the information from the log message. Because indexing and storage are the two sides of our study, we have separated the storage into a Hadoop database. We have stressed the key role of our Spark components, one per data source. The partition management is the key concept for improving the performance of the Spark SQL component. The data storage into data frames during the micro batches is particularly suitable for the management of flows originating from Kafka files. We observed that our approach supported a large volume of logs.

From the filtered logs, we presented the construction of our SVM model based on work from the Center for Pattern Recognition and Data Analytics, Deakin University, (Australia). We were thus able to classify the recognized log patterns into classes of anomalies. This means that we can identify the associated maintenance operations. Finally, to measure the impact of our distributed analysis system, we wanted to build automatically reports based on templates and highlight indexing and storage activity. Our study also shows the limits that we want to push back, such as the management of log patterns. The use of an AI model is not the guarantee of an optimal result. We want to make more.

A first perspective will be to improve the indexing process based on a custom schema. We think that the use of DisMax query parser could be more suitable in log requests where messages are simple structured sentences. The similarity detection makes DisMax the appropriate query parser for short structured messages. The log format has a deep impact on the Solr schema definition and about the anomaly detection. We are going to evolve our approach. In the future, we want to extract dynamically the log format instead of the use of a static definition. We think also about malicious messages, which can perturb the indexing process and introduce bad request in our prediction step. The challenge needs to manage a set of malicious patterns and the quarantine of some message sequences.

A second perspective on the performance comparison obtained with the MongoDB cluster with replication and a sharded MongoDB cluster or horizontal scaling. In that context, data are distributed across many MongoDB servers. The main purpose of sharded MongoDB is to scale reads and writes along multiple shards and then to reduce the communication time.

References

[1] F. Mourlin, G. L. Djiken and N. Laurent, “Big Data for Monitoring Mobile Applications,” The 8th International Conference on Big Data, Small Data, Linked Data and Open Data, ALLDATA, IARIA. April 2022.

[2] A. Oliner, A. Ganapathi, and W. Xu, “Advances and challenges in log analysis,” Communications of the ACM, 2012, 2nd ed., vol. 55, pp. 55-61.

[3] T. F. Yen et al., “Beehive: Large-scale log analysis for detecting suspicious activity in enterprise networks,” In Proceedings of the 29th Annual Computer Security Applications Conference, pp. 199-208, December 2013.

[4] S. He et al., “Experience report: System log analysis for anomaly detection,” In 2016 IEEE 27th International Symposium on Software Reliability Engineering (ISSRE), pp. 207-218, October 2016.

[5] B. Debnath et al., “Loglens: A real-time log analysis system,” In 2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS), pp. 1052-1062, July 2018.

[6] Q. Cao, Y. Qiao, and Z. Lyu, “Machine learning to detect anomalies in web log analysis,” In 2017 3rd IEEE International Conference on Computer and Communications (ICCC), pp. 519-523, December 2017.

[7] W. Li, “Automatic log analysis using machine learning: awesome automatic log analysis version 2.0.,” 3 edition, November 2013.

[8] A. Juvonen, T. Sipola, and T. Hämäläinen, “Online anomaly detection using dimensionality reduction techniques for HTTP log analysis,” Computer Networks 91, pp. 46-56, November 2015.

[9] J. Dongo, C. Mahmoudi, and F. Mourlin, “Ndn log analysis using big data techniques: Nfd performance assessment,” In 2018 IEEE Fourth International Conference on Big Data Computing Service and Applications (BigDataService), pp. 169-175, March 2018.

[10] S. Melink et al., “Building a distributed full-text index for the web,” ACM Transactions on Information Systems (TOIS), 2001, vol.19, n°3, pp. 217-241.

[11] J. He, H. Yan, and T. Suel, “Compact full-text indexing of versioned document collections,” In Proceedings of the 18th ACM conference on Information and knowledge management, pp. 415-424, November 2009.

[12] Q. M. Abdul, S. Cheng, and V. Hristidis, “A comparative study of secondary indexing techniques in LSM-based NoSQL databases,” In Proceedings of the 2018 International Conference on Management of Data, pp. 551-566, May 2018.

[13] W. Luo et al., “Guidelines for developing and reporting machine learning predictive models in biomedical research: a multidisciplinary view,” Journal of medical Internet research, 12 ed., vol. 18, 2016.

[14] P. Henderson et al., “Towards the systematic reporting of the energy and carbon footprints of machine learning,” Journal of Machine Learning Research, 2020, 248 ed., vol. 21, pp. 1-43.

[15] L. M. Stevens et al., “Recommendations for reporting machine learning analyses in clinical research. Circulation: Cardiovascular Quality and Outcomes,” 2020, 10 ed., vol. 13.

[16] D. P. Dos Santos and B. Baeßler, “Big data, artificial intelligence, and structured reporting,” European radiology experimental, 2018, 1st ed., vol. 2, pp. 1-5.

[17] S. Afonin, A. Kozitsyn, and I. Astapov, “SQLReports yet another relational database reporting system,” In 2014 9th International Conference on Software Engineering and Applications (ICSOFT-EA) IEEE, pp. 529-534, August 2014.

[18] K. Koitzsch, “Advanced Search Techniques with Hadoop, Lucene, and Solr,” Pro Hadoop Data Analytics, Apress, Berkeley, CA, 2017, pp. 91-136.

[19] J. Kumar, “Apache Solr search patterns,” Packt Publishing Ltd, 2015.

[20] M. F. Ghalwash, D. Ramljak, and Z. Obradović, “Early classification of multivariate time series using a hybrid HMM/SVM model,” 2012 IEEE International Conference on Bioinformatics and Biomedicine, IEEE, pp. 1-6, 2012.

[21] M. Assefi, E. Behravesh, G. Liu, and A. P. Tafti, “Big data machine learning using apache Spark MLlib,” 2017 IEEE International Conference on Big Data (Big Data), 2017, pp. 3492-3498.

[22] T. D. Nguyen, V. Nguyen, T. Le, and D. Phung, “Distributed data augmented support vector machine on Spark,” 23rd International Conference on Pattern Recognition (ICPR), IEEE, 2016.

[23] F. Gosewehr et al., “Apache camel based implementation of an industrial middleware solution,” In 2018 IEEE Industrial Cyber-Physical Systems (ICPS), IEEE, pp. 523-528, May 2018.

Jurassic Park 2.0: A Reconfigurable Digital Twins Platform for Industrial Internet of Things

Eliseu Pereira, Gil Gonçalves
SYSTEC - Research Center for Systems and Technologies, Faculty of Engineering, University of Porto
Rua Dr. Roberto Frias, 4200-465 Porto, Portugal
Email: {eliseu, gil}@fe.up.pt

Abstract—**Digital Twin (DT) emerged as an Industry 4.0 concept that reflects the behavior of physical entities and processes in the digital environment providing real-time information and insights. Typically, DTs are deployed on specific scenarios or components, virtualizing services and monitoring process variables. This extremely focused software implementation makes harder the adaptation and replication of DTs for new components with similar characteristics. This work focuses on upgrading an existent Internet of Things (IoT) platform, Jurassic Park, to permit the development of flexible DTs, facilitating the implementation and modification of IEC-61499 compliant Cyber-Physical Systems (CPS). The upgraded Jurassic Park is a web-based platform that manages IEC-61499 compatible devices executing the runtime environment (DINASORE). The platform was validated in 2 different scenarios, 1) in a distributed system composed of different DTs and 2) controlling a gripper of a robotic arm.**

Index Terms*—*Cyber-Physical Systems; Digital Twin; IEC-61499; Industrial Internet of Things; Virtualization.

I. Introduction

The changeover to Industry 4.0 was introduced in the shop-floor of new information and communication technologies such as the Industrial Internet of Things (IIoT), Cyber-Physical Systems (CPS), among others [1]. The digitalization and virtualization of the physical devices provided to users and operators support for communicating with other devices, tracking errors, optimizing production and many other advantages [2]. Digital Twins (DTs) implement virtual models of physical entities to mirror the geometry and behavior of those entities in the digital domain [3]. With those digital models, real-time monitoring and control of the physical entities can be achieved [4]. Cyber-Physical Production Systems (CPPS) need to be easily reconfigurable since these systems stand out by responding quickly to changes in production and being flexible enough to introduce new functionalities according to the different needs of the production lines. One of the existent standards for the CPPS reconfiguration is the International Electro-technical Commission (IEC) 61499 standard [5], which allows the encapsulation of different functionalities in software modules, the so-called Function Blocks (FBs).

Several DTs proposals have been made recently; however, there is a lack of flexible solutions that are applicable in the production lines [6]. Nowadays, the creation of a DT is a centralized process oriented toward the device intended to be reflected in the digital world. This makes the DT not very flexible. In a complex system with diverse physical entities fulfilling different functionalities, reusing a DT is a critical process and will require increased computational and human effort. This limitation could be solved by using standards for the deployment of reconfigurable CPPS, such as the IEC-61499 standard. However, its integration with the DT concept is still limited and barely explored by the scientific community. Some of the most relevant limitations are the difficulty of managing complex systems with many software modules and the lower flexibility and adaptability of the systems to new tools at the software and hardware level [7].

The main objective of this work was to upgrade an IoT platform [1][8], Jurassic Park, to enable the development of flexible DT applications integrated with the physical devices (executing the DINASORE [9] runtime environment). The DT solution will include 1) a monitoring module, 2) a control module, and 3) a visualization module. The monitoring module is intended to be the processing component of the information that arrives from the physical environment. The main purpose of this module is to understand the behavior of physical entities and store that information in the digital world. The DT control module aims to be the response feature of the DT solution. Through this module, the user can act on the physical entities, thus making the system an analysis component and acting component with the purpose of modifying the system. The general idea behind the DT visualization module is to allow the management of the DTs from the application system. The solution was validated in 2 different scenarios, 1) to monitor and control a distributed CPS, and 2) to control a robotic arm gripper.

The remainder of the paper is structured as follows: in Section II, we present the literature review, Section III details the architecture of the developed solution, Section IV exposes the implementation carried out and Section V presents the main tests and results made to validate the DT solution developed. The discussion of upgraded platform characteristics is detailed in Section VI, and the conclusions and future work are presented in Section VII.

II. Literature Review

Digital Twins (DTs) can adopt different methodologies, concepts, and technologies, providing simulation, monitoring, or optimization capabilities. For that, we initially present the main technologies used as the basis for implementing the DT

TABLE I: Key enabling technologies for DTs

Tools for the Physical World	Tools for DT modeling	Tools for DT data	Tools for DT services	Tools for connections in DTs
VisionPro, Predix, ROS, Matlab, other software	Simulink, Stella, Ansys Twin Builder other software	MySQL, MongoDB, Beacon, other software	Simulink , Labview, Azure IoT, Matlab, other software	Siemens' MindSphere, Predix, other software

platform, and after the related work presents a description of the more relevant DT implementations.

A. Background

Several enabling technologies allow the implementation of CPSs [10], for different purposes, like device management, data collection, workflow orchestration, among others. The DT solution stacks different technologies, having as a basis an environment with essential capabilities, like reconfiguration of software [11], data interoperability, or communication transparency [12].

The DINASORE [9] is a distributed platform that enables the pre-processing of data in edge devices using Python-based FBs. This platform allows the construction of a wide variety of FBs by the user, with different goals, like controlling drivers, integrating sensors, and applying processing techniques, among other functionalities. The DINASORE uses the 4DIAC-IDE as a graphical user interface to orchestrate the FB pipelines deployed in the distributed CPS.

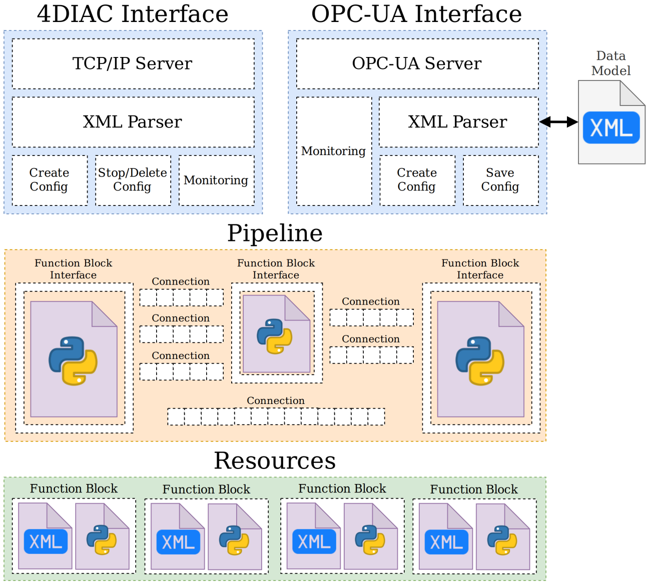


Fig. 1: DINASORE architecture.

The 4DIAC-IDE [13] is a tool developed by Eclipse that allows the orchestration of a workflow of FBs, after being deployed in a distributed system. With this tool, it is possible to draw the FBs distributed systems through a graphical interface and then deploy it into the devices connected to the network.

The Jurassic Park [8], in Figure 2, is a web-based application that acts as a centralized repository of FBs, allowing the creation and management of FBs. Jurassic Park comprises two components: the backend and the web application. This application is integrated with the DINASORE and the 4DIAC-IDE, allowing the complete deployment and management of distributed systems through the IEC-61499 standard.

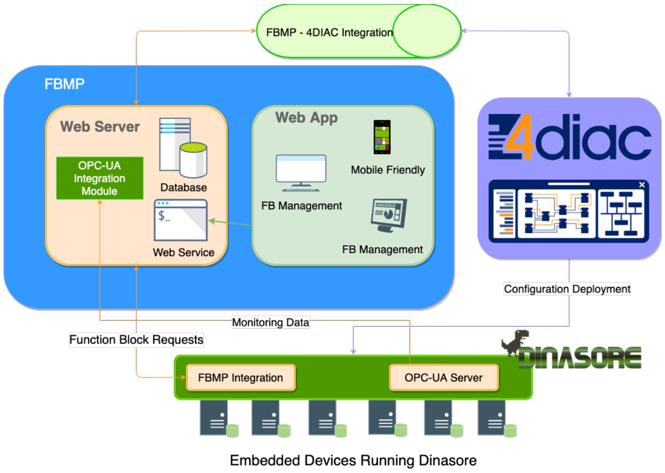
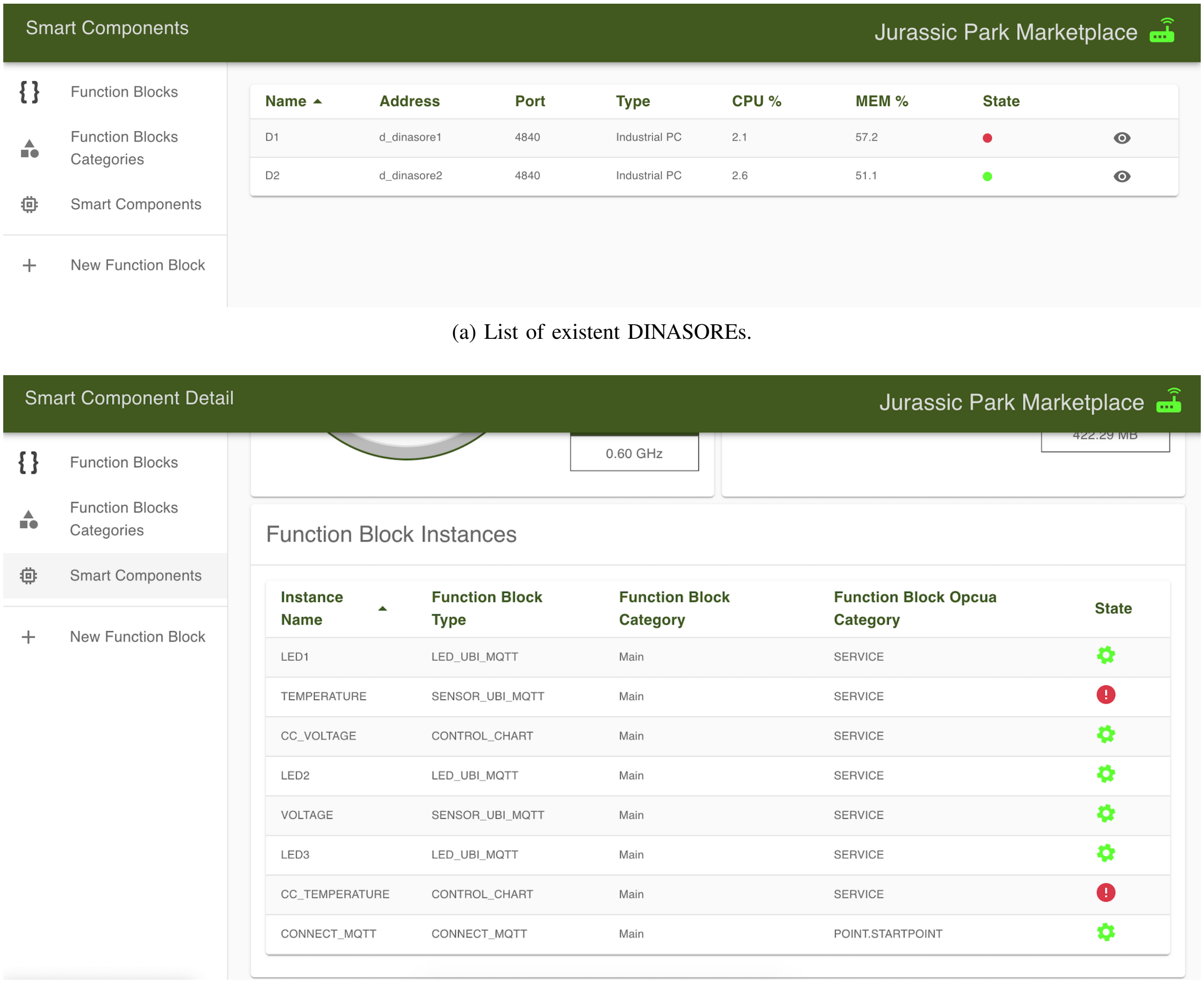


Fig. 2: Original Jurassic Park architecture.

The Web Server contains the application and database servers and should run on the machine where the 4DIAC-IDE is running. The Web Server runs a Node Application that exposes an API composed of Web Services to manage the creation, edition, deletion, and retrieval of the FBs. This application also functions as an OPC-UA client connected to the different OPC-UA Servers running on the embedded devices so it can track them in real time and serve this information to the Web App through a TCP WebSocket connection. The Web Server also runs an FTP Application to serve the needed files to the different embedded systems running on the network. The Web Server runs on the machine where the instance of the 4DIAC-IDE is running, as this integration is made through the machine file system. The Backend also comprises an instance of a MySQL Server Database, which stores all the meta-information related to the software modules.

The Web App used by the Jurassic Park administrators and all the users needs to track the devices in real-time. This application allows, in a friendly way, all the operations required to manage the function blocks (creation, edition, deletion, and retrieval of modules). These operations are accomplished by calling the web services exposed by the Web Server, specifically in the Node Application. The CPS devices are running the DINASORE RTE, which integrates with Jurassic Park using a communication bridge implemented in HTTP and FTP.



Name	Address	Port	Type	CPU %	MEM %	State	
D1	d_dinasore1	4840	Industrial PC	2.1	57.2	🔴	👁
D2	d_dinasore2	4840	Industrial PC	2.6	51.1	🟢	👁

(a) List of existent DINASOREs.

Function Block Instances

Instance Name	Function Block Type	Function Block Category	Function Block Opcua Category	State
LED1	LED_UBI_MQTT	Main	SERVICE	⚙
TEMPERATURE	SENSOR_UBI_MQTT	Main	SERVICE	!
CC_VOLTAGE	CONTROL_CHART	Main	SERVICE	⚙
LED2	LED_UBI_MQTT	Main	SERVICE	⚙
VOLTAGE	SENSOR_UBI_MQTT	Main	SERVICE	⚙
LED3	LED_UBI_MQTT	Main	SERVICE	⚙
CC_TEMPERATURE	CONTROL_CHART	Main	SERVICE	!
CONNECT_MQTT	CONNECT_MQTT	Main	POINT.STARTPOINT	⚙

(b) DINASORE function blocks state.

Fig. 3: Jurassic Park graphical user interface.

B. Related Work

As identified in [14], currently, there is a huge need to provide on-demand manufacturing services through Industrial Internet of Things (IIoT) networks. These services are easily achieved with the implementation of cloud-based manufacturing solutions. The literature review presented in [15] shows a set of cloud solutions available to help in the implementation of Digital Twins (DTs), in particular, the Microsoft Azure IoT solution and the Amazon Web Services (AWS) IoT solution. When large industrial companies began to realize that the implementation and consequent application of DTs could significantly impact the efficiency of their production lines, they began their initiative to develop their tools [16].

In [17], the authors present DT use cases in the industrial environment, where Siemens presents two DT implementations, one for the power system and the other for the wastewater plant. General Electric has not only developed a DT for a wind farm. Still, this implementation has also proved that a wind farm should be operated, developed, and maintained more efficiently. British Petroleum has developed a DT of oil and gas facilities in more restricted areas. The air vehicle manufacturer Airbus has used DT-based solutions to monitor its production lines and to optimize its operations. Going into a more specific example of a DT implementation in the industry context, [18] presents a DT within a hollow glass manufacturing production line. The authors of this implementation argue that this work has allowed them to realize that digital simulations greatly impact how production lines can be optimized since they reflect their behavior in the real world.

The authors in [19] present a DT solution in the commercial greenhouse sector. The article highlights the need for commercial greenhouse production systems to become more energy-efficient while maintaining sustainable production. Therefore, the DT solution mentioned was proposed to provide a control and monitoring feature of the production flow of a greenhouse production system. The authors of this article also defend two essential ideas. One of them is that, with DT implementations, it is possible to turn the industrial environment into more energy and climate-friendly, leading to reduced costs within the production lines. The other idea is that, nowadays, the industrial environment requires developed generic DT frameworks capable of controlling the processes accurately and responding to changes in the orders of the production lines.

III. Architecture

The DT platform was conceptualized to structure multiple DTs, allowing their monitorization and control. This concept is composed of three main objects, 1) the digital twin, which is at the top of the architecture and is responsible for controlling the functionalities of the physical equipment, 2) the functionality, which describes the DT characteristics, mainly the monitoring system variables and the functions to trigger the execution of specific actions, and 3) the device, that reflects the physical entity in the digital world, like sensors, machines, etc. The DT platform developed is embedded in Jurassic Park, taking advantage of some of its features. Figure 4 shows the DT proposed solution composed of three components, the DT visualization module, the DT monitoring module, and the DT control module.

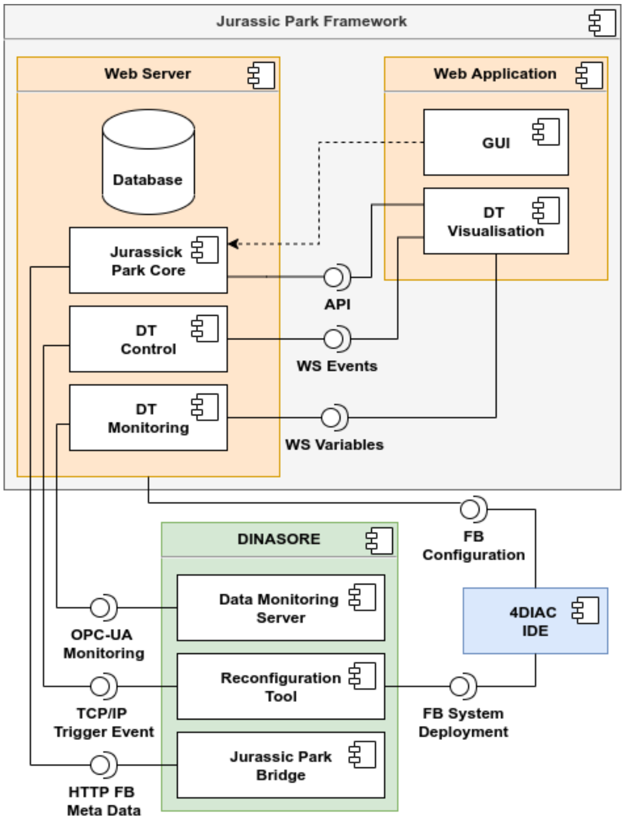


Fig. 4: DT architecture integrated with Jurassic Park and DINASORE.

The DT visualization component (presented in Section IV-C) allows the interaction between the DT platform and the user. This component is incorporated into the web application of Jurassic Park. The main functionalities of this visualization component focus on the management of the DTs. Additionally, this component will be responsible for triggering the monitoring and control requests from the DTs. The DT monitoring module (presented in Section IV-D) is intended to serve as a processing tool for the information that arrives from the physical entity to the digital one and serves the information to the user through the graphical interface. The component is incorporated into the web server of Jurassic Park using the Open Platform Communications - Unified Architecture (OPC-UA) communication protocol as a data source to collect information from the physical world. The monitoring module collects data from the FBs variables and events running on the DINASOREs connected to the platform, allowing the user to filter the ones to monitor. The DT control component (presented in Section IV-E) permits triggering functionalities over the physical entities connected to the platform. The component uses the variables and events from the FBs on the devices connected to the platform and assigns them a trigger feature. In this way, the user will have the opportunity to manipulate the events of the FBs. As in the case of the monitoring component, the control component is also incorporated into the web server.

The backend component (webserver) is a complex module that uses several technologies to consolidate the integration of all the services it supports. This component is responsible for the HTTP API that manages the FBs and the real-time communication of the devices connected to the platform and the database. One of the most relevant technologies used in the backend is the Node.js framework, which integrates several backend components such as the API, the real-time communication with the web application, and the real-time communication with the DINASOREs, through the OPC-UA communication protocol. The backend component uses MySQL as database. The backend also includes a File Transfer Protocol (FTP) server that allows the download of FBs by the DINASOREs.

The frontend component (web application) is the platform's graphical user interface, and it was developed using React. This component is responsible for the page components (buttons, boxes, tables, among other User Interface (UI) elements), which are customized according to the existent resources in the CPS. The automatic update of the information is performed using the Socket IO library, which supports real-time communication with the backend. It avoids the need to refresh the web page to update the information.

IV. Implementation

The upgrade of the Jurassic Park platform followed an agile methodology of development using git that enabled the usage of stable versions for deployments while the platform was upgraded. As follows, this section presents the main Jurassic Park components as well as the top upgrades, mainly the Digital Twin features.

A. Jurassic Park Backend

As mentioned before, Jurassic Park's backend is divided into several components, described in Figure 5. The API component uses a JavaScript package to handle and correctly route the HTTP requests: Express. This component is a composition of other components called API modules. Each of these modules handles a list of requests, and the set of all modules forms the full HTTP Rest API. The following items list the available API operations:

- **GET /function-block/**: Returns a list of the FBs available in Jurassic Park.
- **GET /function-block/:type**: Endpoint responsible for giving the information of the FBs to the DINASOREs connected.
- **POST /function-block/**: Endpoint responsible for creating a new FB in Jurassic Park.
- **PUT /function-block/:id**: Endpoint responsible for editing a specific FB identified by the FB id.
- **DELETE /function-block/:id**: Endpoint responsible for deleting a specific FB identified by the FB id.
- **GET /function-block-category/**: Returns a list of the FB categories available in the platform.
- **POST /function-block-category/**: Endpoint responsible for creating a new FB category in Jurassic Park.
- **PUT /function-block-category/:id**: Endpoint responsible for editing a specific FB identified by the FB category id.
- **DELETE /function-block-category/:id**: Endpoint responsible for deleting a specific FB category identified by the FB category id.
- **POST /smart-component/**: Endpoint responsible by the DINASORE to announce the connection to Jurassic Park.

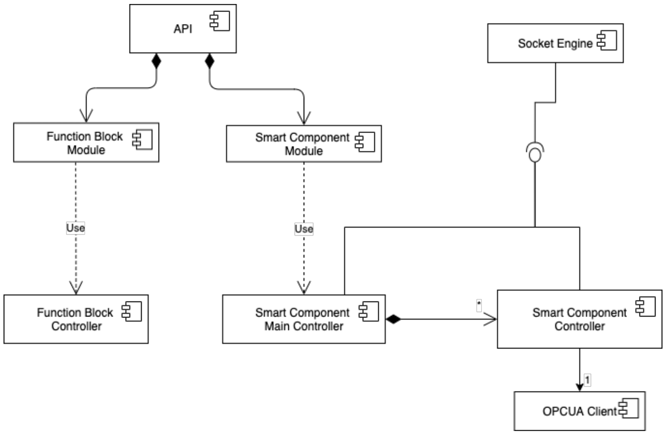

API
Socket Engine
Function Block Module
Smart Component Module
Use
Use
Function Block Controller
Smart Component Main Controller
*
Smart Component Controller
1
OPCUA Client

Fig. 5: Jurassic Park backend components diagram.

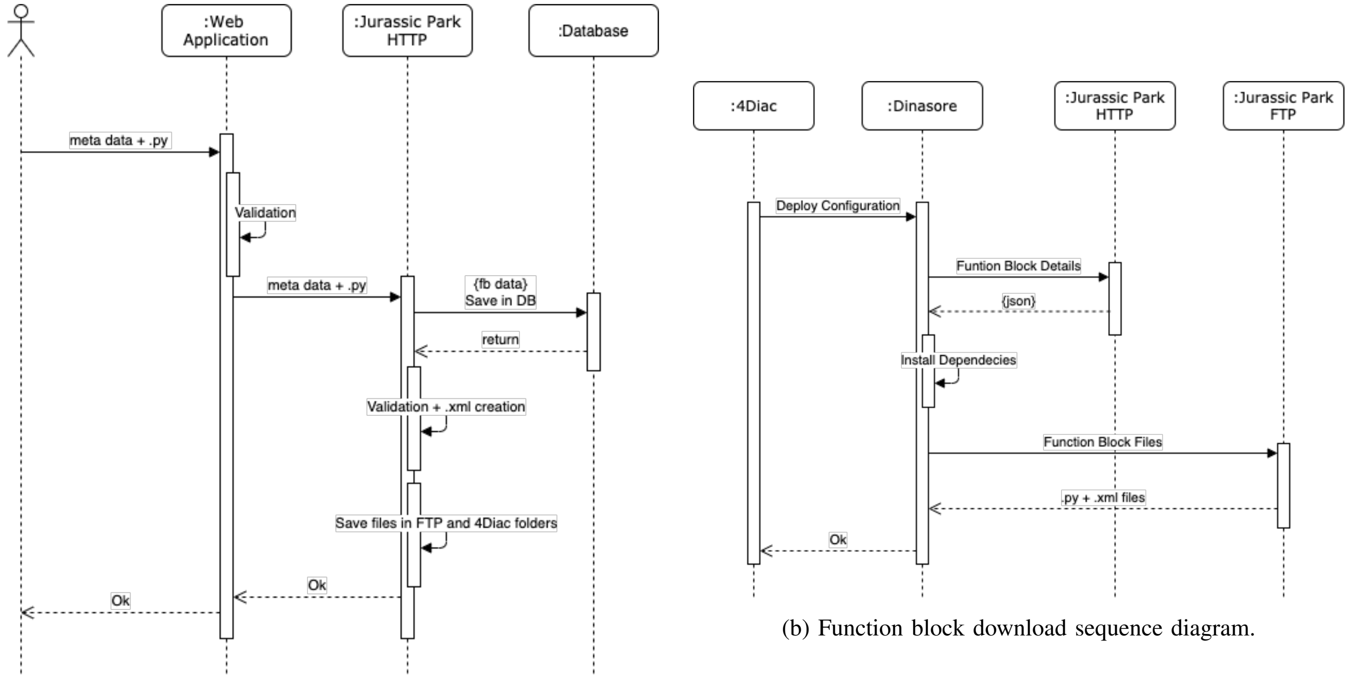
To establish real-time communication between the Web Application running in a browser and the Jurassic Park backend, there must be a permanent communication channel between these two entities. To achieve this, the approach was to use WebSocket technology. The Socket Engine component provides an interface used by both the Smart Component Main Controller and the Smart Component Controller to send data to the connected sockets functioning as a bridge between the controllers and the client connected and running the Web Application on a browser.

The Function Block Controller and the Smart Component Controller were developed to control a specific application area. The first one controls the CRUD Operations of the FBs and FB Categories, and the second controller, a more complex one, controls the DINASORE state at any given moment. The Function Block Controller was developed as a singleton class with methods to create, update, delete and retrieve function blocks and categories. The Smart Component Controller is a singleton class like the Function Block Controller. However, this controller will contain a list of Smart Component Controllers (sub-components), one for each connected smart component device.

B. Jurassic Park Frontend

As previously mentioned, the Jurassic Park frontend uses React to build the web application. It is a JavaScript library developed by Facebook that switches from the imperative programming style to a declarative one. This way, the application becomes much more modular, allowing the reusage of some components so they can act as a sort of template. The web application structure has been divided into the following categories; each category is a folder containing other subfolders or the actual implementation file in React.

- **Components**: This category contains the basic UI components of the application; these components were subdivided into the Function Block folder, with all the UI elements to manage the Function Blocks, the Smart Components folder, to monitor the Smart Components connected to the network, the Function Block Categories folder, containing the UI elements to allow the users to manage the Function Block Categories and the templates folder, containing the UI elements meant to be reused in the other components like charts, navigators and tables.
- **Services**: This category of files was created to accommodate the services needed to communicate with external system components, in this case, with the Jurassic Park backend. Inside this category, there are two subcategories; the HTTP module, with all the functions to fetch and send data to the backend; and the Socket module, with functions to communicate via TCP sockets with the backend, uses Socket IO external package.
- **State**: The paradigm of declarative programming and React, in particular, is straightforward and based on a UI change in reaction to an internal change in the component state. This state can be a simple variable like a string, number, array, or even a set of these. Once changed, this state triggers an update in all the UI elements dependent on that state. However, for complex applications with more than one component, there should be a place to centralize all the state variables used by different components. Therefore, the functions built inside this module can extract or change the state, so any component that needs to change this centralized state can only do that



(a) Function block creation sequence diagram.

(b) Function block download sequence diagram.

Fig. 6: Sequence diagrams for the creation (a) and download (b) of function blocks.

by using this API. For example, the Function Block categories fetched from Jurassic Park backend should be centralized here so the Function Block components and the Function Block Categories components can use them to build their respective UI.

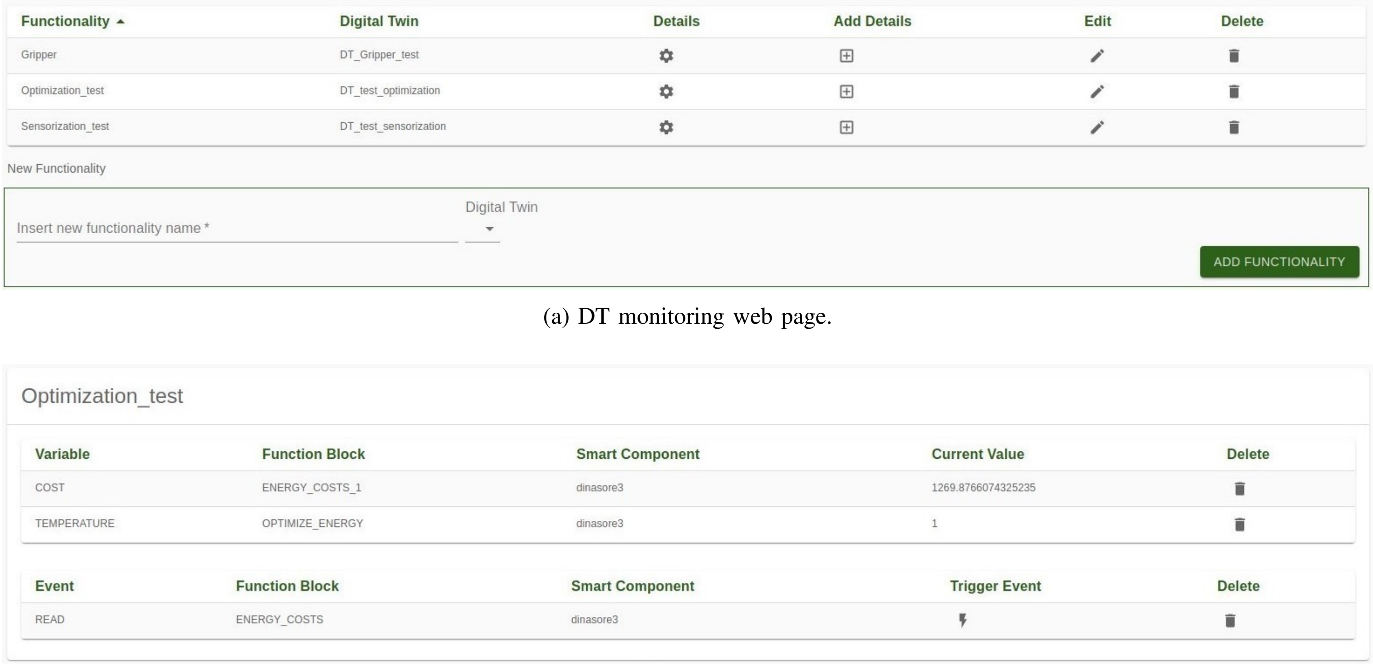
Using these categories as a basis, Jurassic Park includes web pages used for interacting with the user, through a browser, on a desktop, or mobile device. The main pages of the application are in the following list, where the first three enable the CRUD operations.

- **Function Block List**: The first page is the list of all the FBs available in the marketplace. In this list, it is possible to see all the main characteristics of each FB, including its type, description, category, and general category.
- **New Function Block**: The second page uses a form to create a new FB. This form contains all the fields for creating a new FB, including all the mechanisms to ensure the correct creation of the FB, ensuring that it follows the correct structure. Those validations include verifying that there can't be more than one FB with the same name or that the user uploaded the FB Python file.
- **Edit Function Block**: The third page allows the edition of a FB. In this page, the FB form had already been built, so it was just a matter of reusing it. The difference is how it is loaded when the form is loaded; the data of an already existing FB is passed on.
- **Smart Component List**: This page lists the DINASOREs connected to the network. This list shows the information on each device that has already established a connection to Jurassic Park. Hence, a DINASORE can be connected or disconnected, as the green or red light indicates in Figure 3a. The list also shows, for each component, its name, address, port, CPU percentage current usage, and memory used.
- **Smart Component Detail**: This page shows the monitored information of a DINASORE in Figure 3b. The monitored information includes the FB executing in that DINASORE, detailing each FB state, where the green icon means the FB is running correctly, and when it goes to red, it stops working. Like the previous component, this page is dynamically updated, making it a live dashboard to monitor the DINASORE.
- **Function Block Category**: This page lists the available FB categories. For each category created, the application shows the category's name and a list of FBs in that category. The categories managed here are available in the FB form.

C. Digital Twin GUI

The DT visualization component allows the user to create and manipulate DTs. The platform allows users to group the devices they want to monitor and control into a category called DT. Subsequently, they can associate that DT with a general functionality. Therefore, to describe the page elements in a more user-focused way, we list below the different pages implemented to fulfill the requirements of the DT component.

The new DT page allows the user to add new DTs with a specific name and to associate them with respective DINASOREs. The page has a menu for creating a new DT; in this menu, the user has access to all the devices communicating



Functionality ▲	Digital Twin	Details	Add Details	Edit	Delete
Gripper	DT_Gripper_test	⚙	⊞	✎	🗑
Optimization_test	DT_test_optimization	⚙	⊞	✎	🗑
Sensorization_test	DT_test_sensorization	⚙	⊞	✎	🗑

New Functionality

Insert new functionality name *

Digital Twin

ADD FUNCTIONALITY

(a) DT monitoring web page.

Optimization_test

Variable	Function Block	Smart Component	Current Value	Delete
COST	ENERGY_COSTS_1	dinasore3	1269.8766074325235	🗑
TEMPERATURE	OPTIMIZE_ENERGY	dinasore3	1	🗑

Event	Function Block	Smart Component	Trigger Event	Delete
READ	ENERGY_COSTS	dinasore3	⚡	🗑

(b) Functionality details web page.

Fig. 7: Digital Twin GUI, including the monitoring (a) and functionalities details (b) web pages.

with Jurassic Park in real-time. Therefore, to create a new DT, the user only has to complete the field to insert the name of the new DT, open the list of available devices, and select those he wants to associate with. It should be noted that the user can choose more than one DINASORE. Additionally, the user can only create the new DT if he has chosen at least one DINASORE.

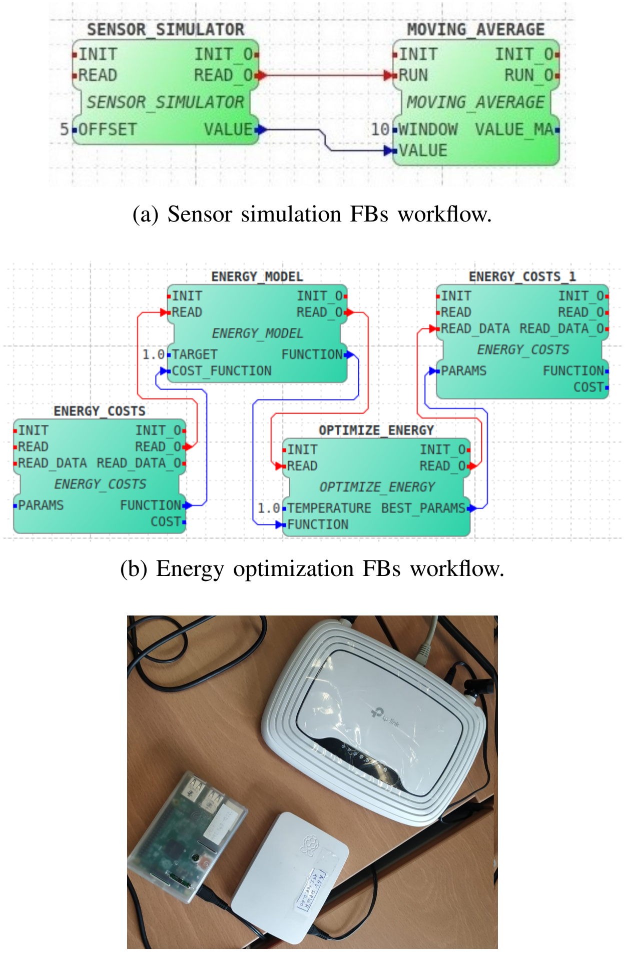
The Digital Twin monitoring page, in Figure 7a, performs all the management of the DT. On this page, the user can observe the currently active functionalities with the respective DT. Besides that, on this page, the user can create other functionalities and associate a DT capable of observing the variables and/or events of interest. Looking first at the feature of the page concerning the creation of a new functionality, the user can choose the name of the functionality they want to add to the monitoring platform. After choosing the name, the user will have to choose one DT from the range of available DTs previously created on the New Digital Twin page, which they want to associate with the new functionality. Through this page, the user will have at his disposal a table, listing all the functionalities currently active and a set of features. The *Details* feature allows the user to view in detail the information collected from the variables and events currently being monitored. The *Add Details* feature enables users to choose the variables and events they want to monitor on a given FB. With the *Edit* feature, the user can edit the name of the functionality. In addition to being able to edit and manipulate the variables and events that each functionality will monitor, the user can also delete the available functionalities they want with the *Delete* feature.

The interface redirects to a new page whenever the user presses the functionality details button. The functionality details page, in Figure 7b, has two different tables, one for monitoring variables and another for triggering events. Note that the page only displays the variables and events that have been previously selected on the DT monitoring page. Thus, the user has information organized either by variables or by events at their disposal. Regarding the variables monitoring, the user can observe in the table some information about the variable, such as the variable name, the FB where the variable is being monitored, the device that is allocating, and the variable's current value.

Additionally, the user can delete the variables they no longer wish to monitor by clicking on the delete button in each row of the table. The event monitoring table does not differ much from the variable monitoring table. It is also possible to see the name of each event selected by the user, the associated FB and the DINASORE where it is being mapped. However, unlike the variable monitoring table, the event table has a button that allows the user to trigger an action on the associated event. When the user presses the trigger event button, Jurassic Park automatically sends a request to the server to execute the event on the associated FB. The user also has a button on the event table that allows them to delete events they no longer wish to observe.

D. Digital Twin Monitoring

The monitoring component of the DT platform is responsible for requesting and collecting information on each DT, which is then made available in the DT visualization component. For that, the user should create the different DTs and functionalities and define the monitored devices, variables, and events. With the previous objects created, the following integration is to automatically receive real-time feedback on the values of the monitored variables. Initially, the request is sent via WebSockets from the web application, where a persistent listener waits for feedback from the monitored variables. On the server side, the DINASORES communicate using an OPC-UA client. When the monitored variables present a change, the OPC-UA client automatically identifies it. Then the data is sent back to the backend controller, which ensures the notification of the new value to the web application, waiting for the information through the listener function. When the user no longer wants to observe the variables, an event to cancel the subscriptions is sent to the OPC-UA client.



(a) Sensor simulation FBs workflow.

(b) Energy optimization FBs workflow.

(c) Physical scenario, including a wireless router and 2 raspberry pis.

Fig. 8: Distributed CPS composition with a sensorization (a) and optimization (b) workflows and the physical scenario (c).

E. Digital Twin Control

The DT control component allows users to trigger events in certain FBs running on specific DINASOREs. After interacting with the UI element, a message containing the triggered event's information is sent to the backend component. Having the information regarding the event in the backend, a function sends the action to the device in question, using Transmission Control Protocol/Internet Protocol (TCP/IP) sockets. For that, the function uses as input the event information, in particular the FB name, event name, device IP address, and port. When the DINASORE receives the message, it pushes an event on the specified FB and executes the triggered functionality.

V. Experiments and Results

The experiments performed to validate the DT component of the platform focused on two use cases that allow testing of the different implemented features. The experimented use cases are 1) the monitorization of simulated sensors and the optimization of energy in a distributed CPS and 2) the manipulation of a robotic arm gripper.

A. Monitoring and optimizing a distributed CPS

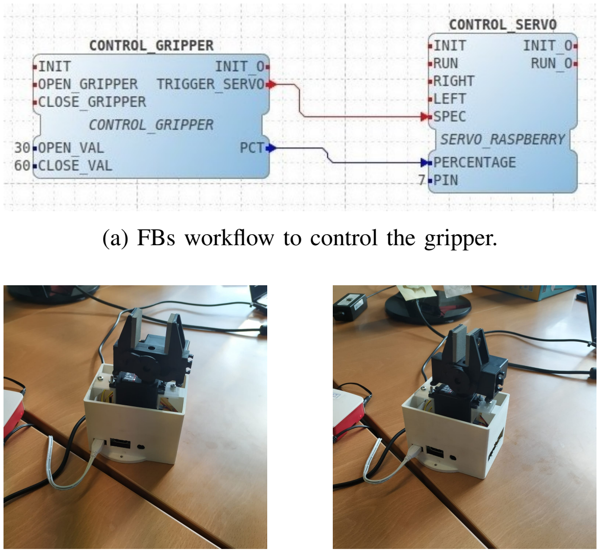
The first validation scenario is a distributed system with different devices connected to the platform. This experiment consisted of connecting a set of raspberry pis to the platform, which are responsible for executing two different FBs pipelines/workflows, one for sensing purposes and the other for energy optimization. This set of FBs pipelines allows the monitoring of variables and trigger of events in a distributed system, allowing the validation of the previously explained monitoring and controlling features of the DT platform.

The first raspberry pi was used to validate the variable monitoring component, using a small FBs workflow that simulates a sensing system, present in Figure 8a. The FBs composing the workflow are 1) the SENSOR_SIMULATOR, responsible for generating a random value to simulate the value of a sensor, and 2) the MOVING_AVERAGE, which calculates the average of the last N values. After the deployment of the workflow, the next step was to create a DT for the sensing component associated with the raspberry pi that executes the DINASORE. Then, we created a functionality that includes the following monitoring variables, 1) the variable VALUE (variable containing the simulated sensor data) from the SENSOR_SIMULATOR FB and, 2) the variable VALUE_MA (variable containing the moving average which is calculated) from the MOVING_AVERAGE FB.

The second raspberry pi hosts a set of FBs that optimize the energy consumption of a process, present in Figure 8b. This FB pipeline allows the validation of the event-triggering process through the DT platform and, consequently, the monitoring of variables linked to that event. The FBs composing the workflow are 1) the ENERGY_COSTS, which specifies the function for energy costs regarding velocity and power, 2) the ENERGY_MODEL, which is the model that allocates the energy consumption, using as input the energy costs, and 3) the OPTIMIZE_ENERGY, which optimizes the given function, through the use of the Dual Annealing algorithm. The DT associated with the optimization workflow includes functionalities that contain the following variables and events: 1) the COST variable (optimized energy costs) present in the ENERGY_COSTS_1 FB, and 2) the READ event (triggering optimization) of the ENERGY_COSTS FB. With this variable and event combination, we can trigger an event via the platform, sent to the raspberry pi, that executes the optimization.

B. Creating a Digital Twin for a Gripper

The second scenario focuses on creating a DT for a robotic arm gripper, validating the physical component's control and monitoring features. The component used was a 3D-printed gripper using a raspberry pi as a controlling system for the servo motor. This experiment consisted in controlling the gripper through the developed DT platform by manipulating the event trigger to make the gripper arm open and close according to the event, as shown in Figure 9b and 9c. The FB workflow that allows the gripper's control, present in Figure 9a, is composed of the FBs: 1) the CONTROL_GRIPPER, responsible for identifying whether there is a request to open or close the gripper, depending on which, the FB sends the corresponding percentage to the output (PCT), and 2) the CONTROL_SERVO, which receives as input the percentage and then updates the general-purpose input/output (GPIO) that controls the servo motor with the corresponding value.



(a) FBs workflow to control the gripper.

(b) Gripper open.

(c) Gripper closed.

Fig. 9: Results obtained with the gripper scenario, including the FBs workflow (a) and the physical gripper (b) and (c).

The DT created intends to monitor and control the gripper manipulation. For this use case, two distinct events and a variable (to evaluate whether the events are being correctly triggered) have been added to the functionality associated with the DT. The selected events to control the gripper were the OPEN_GRIPPER and CLOSE_GRIPPER events. To check whether the control on the DT platform is functional, the gripper would have to open when the user pressed the release button. If the close button were pressed, the gripper would have to execute the closing movement. The execution of both movements was validated visually and also through the monitoring variable (percentage), which confirms if the gripper moves according to the monitored percentage.

VI. Discussion

The discussion of the results obtained in the different scenarios will focus on the verification of the upgraded Jurassic Park platform in terms of compliance with IoT requirements [2]. The IoT requirements enable developers and researchers to validate if the IoT solutions comply with current standards and align with modern CPS's goals. The upgraded Jurassic Park platform passed through the evaluation in 5 different requirements, i.e., scalability, reliability, interoperability, timing, and reconfigurability.

- **Scalability**: The platform presents scalability in the number of devices connected, DT representations, monitored variables, and CPPS functionalities.
- **Reliability**: In terms of reliability, the platform was validated in different scenarios, replicating conditions from a real CPS, like distributed communication, execution of devices with low computational resources, and actuation into physical assets.
- **Interoperability**: The interoperability of a CPS enables transparent and fluid communication across distributed devices. The validation scenario with distributed devices (2 raspberry pis) enabled the validation of that requirement.
- **Timing**: During the platform execution, the response times of the variables monitorization and functionalities triggering are near real-time, i.e., approximately between 1 and 2 seconds of response time.
- **Reconfigurability**: As described before, the upgraded Jurassic Park platform enables the quick and easy modification of any parameter of the CPPS, e.g., the monitored FBs or the devices connected to the platform. The GUI has particular pages for the modification

With these findings, the upgraded Jurassic Park platform presents a large set of characteristics and enablers to their deployment in a real CPS. Those characteristics and achieved requirements prove not only the capacity for the platform to execute in a CPS but also facilitate the day-to-day life of the CPS managers and industry operators because it makes easier the traceability of the processes and makes the systems more transparent to humans.

VII. Conclusion

In conclusion, the main objectives the Jurassic Park 2.0 were to upgrade it by implementing a flexible and reconfigurable DT solution capable of increasing the monitoring capacity and enabling the remote control of a CPS. The components developed and upgraded, mainly the DT visualization component, the DT monitoring component, and the DT control component, permitted the CPS to accomplish the described attributes, like DT reconfiguration. The solution was integrated with a mature stack of technologies, including DINASORE. The experiments were performed to support the usability and flexibility of the web-based platform. Finally, it is essential to highlight that one of the most significant contributions of this project was to develop a platform that, given its flexibility and configurability, can be easily integrated into the industrial sector and supports the IEC-61499 standard.

As future work, one of the main goals will be the storage of data generated by the DT variables, considering data volume constraints in terms of storage and data flow/rate. On top of this unstructured database, it will be possible to implement predictive algorithms to forecast and optimize [20][21] the behavior of DTs. Additionally, the entire platform will be integrated into an industrial scenario composed of different machines, including the integration of sensors, simulation of processes, and optimization of resources.

ACKNOWLEDGMENT

INDTECH 4.0 - New technologies for intelligent manufacturing. Support on behalf of IS for Technological Research and Development (SI a Investigacao e Desenvolvimento Tecnologico). POCI-01-0247-FEDER-026653

The authors thank the students involved for helping in the development of the platform, in particular Maria Arieiro.

REFERENCES

[1] E. Pereira, M. Arieiro, and G. Gonçalves, "Reconfigurable digital twins for an industrial internet of things platform," in *INTELLI 2022, The Eleventh International Conference on Intelligent Systems and Applications*, 2022, pp. 30–35.

[2] L. Antão, R. Pinto, J. Reis, and G. Gonçalves, "Requirements for testing and validating the industrial internet of things," in *2018 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW)*, 2018, pp. 110–115.

[3] C. Cimino, E. Negri, and L. Fumagalli, "Review of digital twin applications in manufacturing," *Computers in Industry*, vol. 113, p. 103130, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0166361519304385

[4] K. Ding, F. T. S. Chan, X. Zhang, G. Zhou, and F. Zhang, "Defining a Digital Twin-based Cyber-Physical Production System for autonomous manufacturing in smart shop floors," *International Journal of Production Research*, vol. 57, no. 20, pp. 6315–6334, 2019.

[5] K. Thramboulidis, "Iec 61499 in factory automation," in *Advances in Computer, Information, and Systems Sciences, and Engineering*, K. Elleithy, T. Sobh, A. Mahmood, M. Iskander, and M. Karim, Eds. Dordrecht: Springer Netherlands, 2006, pp. 115–124.

[6] Y. Fan, J. Yang, J. Chen, P. Hu, X. Wang, J. Xu, and B. Zhou, "A digital-twin visualized architecture for flexible manufacturing system," *Journal of Manufacturing Systems*, vol. 60, pp. 176–201, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S027861252100114X

[7] G. Lyu and R. W. Brennan, "Towards IEC 61499-Based Distributed Intelligent Automation: A Literature Review," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 4, pp. 2295–2306, 2021.

[8] J. Pedro Furriel, E. Pereira, J. Reis, and G. Gonçalves, "Jurassic park - a centralized software modules repository for iot devices," in *2021 10th Mediterranean Conference on Embedded Computing (MECO)*, 2021, pp. 1–4.

[9] E. Pereira, J. Reis, and G. Gonçalves, "Dinasore: A dynamic intelligent reconfiguration tool for cyber-physical production systems," in *Eclipse Conference on Security, Artificial Intelligence, and Modeling for the Next Generation Internet of Things (Eclipse SAM IoT)*, 2020, pp. 63–71.

[10] Q. Qi, F. Tao, T. Hu, N. Anwer, A. Liu, Y. Wei, L. Wang, and A. Nee, "Enabling technologies and tools for digital twin," *Journal of Manufacturing Systems*, vol. 58, pp. 3–21, 2021, digital Twin towards Smart Manufacturing and Industry 4.0.

[11] C. Zhang, W. Xu, J. Liu, Z. Liu, Z. Zhou, and D. T. Pham, "A reconfigurable modeling approach for digital twin-based manufacturing system," *Procedia CIRP*, vol. 83, pp. 118–125, 2019, 11th CIRP Conference on Industrial Product-Service Systems. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2212827119304469

[12] E. Pereira, R. Pinto, J. Reis, and G. Gonçalves, "Mqtt-rd: A mqtt based resource discovery for machine to machine communication," in *Proceedings of the 4th International Conference on Internet of Things, Big Data and Security - IoTBDS,*, INSTICC. SciTePress, 2019, pp. 115–124.

[13] A. Zoitl, T. Strasser, and A. Valentini, "Open source initiatives as basis for the establishment of new technologies in industrial automation: 4diac a case study," in *2010 IEEE International Symposium on Industrial Electronics*, 2010, pp. 3817–3819.

[14] Y. Lu and X. Xu, "Cloud-based manufacturing equipment and big data analytics to enable on-demand manufacturing services," *Robotics and Computer-Integrated Manufacturing*, vol. 57, pp. 92–102, 2019.

[15] T. Catarci, D. Firmani, F. Leotta, F. Mandreoli, M. Mecella, and F. Sapio, "A conceptual architecture and model for smart manufacturing relying on service-based digital twins," in *Proceedings - 2019 IEEE International Conference on Web Services, ICWS 2019 - Part of the 2019 IEEE World Congress on Services*. IEEE, Piscataway, NJ, USA, 2019, pp. 229–236.

[16] A. Fuller, Z. Fan, C. Day, and C. Barlow, "Digital Twin: Enabling Technologies, Challenges and Open Research," *IEEE Access*, vol. 8, pp. 108 952–108 971, 2020.

[17] F. Tao, H. Zhang, A. Liu, and A. Y. Nee, "Digital Twin in Industry: State-of-the-Art," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 4, pp. 2405–2415, 2019.

[18] H. Zhang, Q. Liu, X. Chen, D. Zhang, and J. Leng, "A Digital Twin-Based Approach for Designing and Multi-Objective Optimization of Hollow Glass Production Line," *IEEE Access*, vol. 5, pp. 26 901–26 911, 2017.

[19] D. Anthony Howard, Z. Ma, J. Mazanti Aaslyng, and B. Nørregaard Jørgensen, "Data architecture for digital twin of commercial greenhouse production," in *2020 RIVF International Conference on Computing and Communication Technologies (RIVF)*, 2020, pp. 1–7.

[20] E. Pereira, J. Reis, G. Gonçalves, L. P. Reis, and A. P. Rocha, "Dutch auction based approach for task/resource allocation," in *Innovations in Mechatronics Engineering*, J. Machado, F. Soares, J. Trojanowska, and S. Yildirim, Eds. Cham: Springer International Publishing, 2022, pp. 322–333.

[21] R. Rossini, G. Prato, D. Conzon, C. Pastrone, E. Pereira, J. Reis, G. Gonçalves, D. Henriques, A. R. Santiago, and A. Ferreira, "Ai environment for predictive maintenance in a manufacturing scenario," in *2021 26th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2021, pp. 1–8.

Truth or Fake? Developing a Taxonomical Framework for the Textual Detection of Online Disinformation

Isabel Bezzaoui, Jonas Fegert and Christof Weinhardt
Information Process Engineering
FZI Research Center for Information Technology
Karlsruhe/Berlin, Germany
bezzaoui@fzi.de, fegert@fzi.de, weinhardt@fzi.de

Abstract — **Disinformation campaigns have become a major threat to democracy and social cohesion. Phenomena like conspiracy theories promote political polarization; they can influence elections and lead people to (self-)damaging or even terrorist behavior. Since social media users and even larger platform operators are currently unready to precisely detect disinformation, new techniques for identifying online disinformation are urgently needed. In this paper, we present the first research insights of DeFaktS, an Information Systems research project, which takes a comprehensive approach to both researching and combating online disinformation with a special focus on enhancing media literacy and trust in explainable AI. Specifically, we demonstrate the first methodological steps towards the training of a machine learning-based system. This will be obtained by introducing the development and preliminary results of a taxonomy to support the labeling of a 'Fake News' dataset.**

Keywords - Fake News; Disinformation Detection; Machine Learning-Based Systems; Taxonomy.

I. INTRODUCTION

Online disinformation is currently regarded as one of the most serious challenges to democracy, journalism, and free expression, increasing the demand for research on the detection of fraudulent content. The present paper seeks to extend the findings of [1], a research project focusing on using explainable AI to understand and combat online disinformation. As the major news source of today, social media channels and online news portals suffer from non-fact-based reporting and opinion dissemination [2]. Spreading virally, disinformation poses a central threat to the political process and social cohesion. Disinformation is defined as false information, spread with the intention to deceive. 'Fake News' is an example of disinformation, which is why, in line with current literature in ICT research, we use these two terms interchangeably [3]. Deceptive information influences elections and tempts people to engage in (self-)damaging or even terrorist behavior. Accordingly, it displays a generally undesirable phenomenon in public information and opinion-forming processes [4,5]. Besides political radicalization [6], vaccination boycotts are increasingly attributed to disinformation campaigns and thereby present a threat to the general health system [7,8]. Therefore, on the one hand, there is a need for a comprehensive understanding of their mechanisms and spread, and on the other hand, based on this, methods to systematically combat them. People are naturally inclined to consume content with which they are familiar (familiarity bias), whose authors are similar to them (similarity bias), or whose statements they basically agree with (confirmation bias). In particular, confirmation bias is a decisive factor in the spread of disinformation [9]. Platforms such as WhatsApp and Telegram play a major role in the spread of disinformation and could take many preventive measures. They generally lack the appropriate approaches for this, since more emotional arousal and dissent lead to more activities on the platform, and in turn generate more revenue in advertising [10, 11]. Even though Twitter, for example, is experimenting with fact checking, these efforts are far from sufficient to limit the spread of 'Fake News' as those services do not operate across platforms. Thus, DeFaktS tries to empower actual users across various platforms to critically question news and social media content. For this purpose, the project will develop an explainable AI artifact for a participation platform that aims to combat online disinformation campaigns and foster critical media literacy among users by informing them about the occurrence of 'Fake News' in a transparent and trustworthy manner. Precisely, the DeFaktS project develops a data pipeline in which (i) messages are extracted by annotators in large quantities from suspicious social media and messenger groups. Based on this corpus, a machine learning-based system (ii) is trained that can recognize factors and stylistic devices characteristic of disinformation, which will be used for (iii) an explainable AI that informs users in a simple and comprehensible way about the occurrence of disinformation.

Machine data analytics remain challenged by the wide variety of stylistic devices utilized in fraudulent messages, which poses a barrier for merely quantitative approaches to the issue [12]. Empirical findings demonstrate that disinformation content is the hardest to detect. This seems reasonable considering that the false class is dispersed and layered over other classes. The deceptive nature of 'Fake News', where the goal is to make the information appear to be a legitimate piece, may help to explain this [13]. Nevertheless, studies on the structure of disinformation indicate that the substance of authentic and deceptive news articles differs significantly [14]. The DeFaktS project seeks to face this challenge in the following way: The development of a taxonomy of online disinformation (TOD) that entails linguistic features and dimensions of disinformation content shall facilitate and ensure the quality of the data labeling process. One of the difficulties in detecting false news is that some terms and expressions are unique to a particular type of event or topic. When a 'Fake News' classifier is trained on fake versus real articles based on a certain event or topic, the classifier learns event-specific features and may not perform well when used to identify deceptive content based on a different type of event. As a result, 'Fake News' classifiers must be generalized to be event-independent [3]. Another challenge is that the majority of datasets are in English, and German-language datasets are rare [15]. Since the spread of disinformation is not bound to language barriers, creating functioning datasets in other languages is crucial. Recent research addresses the opportunities of different detection methods and their underlying theories [16-19]. What is lacking, however, is a fundamental empirical overview of concrete detection cues supporting the creation of labels for annotating datasets. Furthermore, even though there are numerous empirical papers presenting disinformation classifiers, they offer no explanations on how these classifiers were trained and how exactly the datasets used for training were labeled [20-23]. Although these explanations are critical to the transparency and traceability of the overall research process and prove that current scientific knowledge is considered in the labeling process of the data, little research has addressed this issue. These observations call for the creation of a taxonomy of online disinformation that encompasses broad and event-independent dimensions and characteristics of disinformation, which is still specific enough to precisely identify and label deceiving content. The systematic coordination of knowledge is an ongoing issue in information systems research [24]. The classification of items helps understanding and analyzing complex settings, and therefore, the creation of taxonomies are crucial for research and development [25]. Furthermore, by using taxonomies, a domain's (e.g., disinformation) knowledge body can be organized and given structure [26]. According to the design science epistemology [27], which also covers descriptive knowledge and prescriptive knowledge, taxonomies are examples of conceptual knowledge. Our final taxonomy will display a design artifact in and for itself that will be demonstrated and evaluated before as well as within the labeling process. After the artifact undergoes iterations, it will be made accessible, and thus extendable, to other researchers through scientific publications or open access services. In this paper, we extend our findings [1] by providing insights into the methodological approach of developing a taxonomical framework for the textual detection of online disinformation as well as an overview of our preliminary results. The paper is structured as follows: Section II will give an introductory overview on the current knowledge base on the efforts of combating disinformation as well as the concepts of critical media literacy and trust. Subsequently, the scientific method and first research activities in the project will be presented in Section III. Thereafter, we provide a short overview of our project's preliminary results in Section IV. Finally, the paper concludes with a summary of the project's research endeavors and an outlook on future work related to the project in Section V.

II. Theoretical Foundation

A. Combating Disinformation Using Machine Learning-Based Systems

The fact that nowadays almost anyone can publish content on the internet not only increases the possibility of social participation – it also creates new opportunities for spreading disinformation and propaganda. The COVID-19 pandemic has already produced a flood of false reports and demonstrated the importance of being able to distinguish reliable information from mis- and disinformation. The war on Ukraine also demands a special confrontation with disinformation distributed by state entities [28]. Currently, research on 'Fake News' detection using machine learning-based systems (MLS) is a rapidly expanding field that spans numerous disciplines, including computer science, social science, psychology, and information systems [29-31]. Synoptically, empirical efforts to detect and combat disinformation can be divided into four categories: data-oriented, feature-oriented, model-oriented and application-oriented [2]. The majority of methods concentrate on extracting multiple features, putting them into classification models, such as naive Bayes, logistic regression, or decision trees, and then selecting the best classifier based on performance [32-35]. What is missing from the previous work, however, are empirical evaluations of when the classifiers are put into practice with real users and of what benefits and impact the presented tools may have. For instance, Guess et al. [36] showed that promoting media literacy can help people judge the authenticity of online content more accurately. Their findings suggest that a lack of critical media literacy is a major factor in why people fall for disinformation. Pennycook and Rand [37] found that susceptibility to 'Fake News' is driven mostly by insufficient critical thinking rather than by partisan bias per se. Thus, in order to counter false news, more critical media competence is needed on the part of users. From this point of view, it seems crucial to investigate the potential of MLS detection tools for promoting critical media literacy among social media users.

Furthermore, previous research has demonstrated the importance of trust for the acceptance and perceived usefulness of ICT tools, and MLS in particular [38,39]. Trust is one of the vital components to fostering active, engaged and informed citizens [40]. Transparency is therefore an important aspect when it comes to dealing with disinformation. In this regard, the challenge of how to positively affect and build trust when developing tools for 'Fake News' detection arises. The implementation of an XAI-approach into the development process seeks to make the system's internal dynamics more transparent, as well as the analysis' conclusions more understandable and hence trustworthy to the user. These observations give rise to the need to examine the effect of XAI (Explainable Artificial Intelligence) elements on user trust and thus acceptance and perceived usefulness of the final tool. In order to fill the two above-mentioned research gaps, we would therefore like to address the following research questions in the DeFaktS project:

How to design an artifact for the detection of online disinformation that helps to foster informed and critical thinking?

i. How does the tool promote critical media literacy by helping users identify disinformation more accurately?

ii. How does the tool's XAI-component assist users to trust the algorithm's assessment?

B. Critical Media Literacy

Disinformation is producing uncertainty in the process of information procurement, endangering the public's ability to make informed decisions [41]. In order to foster a critical comprehension of both manipulative communications and the internet as a distribution medium, users must have broad knowledge and a deeper understanding of social media functionalities [42]. Critical media literacy encourages people to consider why a message was sent and where it came from [43]. Following [44], critical media literacy entails developing skills in analyzing media codes and conventions, and the ability to critique stereotypes, dominant values, and ideologies, as well as the competence to interpret media texts' multiple meanings and messages. Furthermore, it assists individuals to use media responsibly, to discern and assess media content, to critically examine media forms, to explore media effects, and based on those abilities to deconstruct alternative media. However, a systematic evaluation of the effects of the usage of MLS 'Fake News' detection tools on the cultivation of critical media literacy is scarce [45]. Schmitt et al. [45] define three dimensions of critical media literacy that can be referred to the critical handling of online disinformation:

1. Awareness: Awareness in this case means to become aware of the existence of disinformation. This includes knowledge of various forms of disinformation (picture, text, or video form, distorted articles, and pseudo media outlets) as well as a deeper understanding of how media, and online media in particular, operate.
2. Reflection: Reflection in the context of critical media literacy is about applying analytical criteria to internet content and determining whether or not it is deceptive. The conscious consideration, reflection, of content with the character of news is relevant, the thorough thinking before an article is liked, shared or the claim of a headline is taken at face value. As a result, reflection utilizes an individual's knowledge, abilities, and attitudes to critically evaluate (media-communicated) information based on specific criteria including credibility, source, and quality.
3. Empowerment: Individuals' confidence in their ability to detect manipulative messages, participate in social discourses, and actively position themselves against disinformation is cultivated through empowerment strategies and methods. In this context, empowerment can be defined as a certain form of behavior that encompasses a person's ability to recognize and express doubts about specific content as well as express their own thoughts.

In the DeFaktS project, these three dimensions will be used to investigate whether and to what extent the developed MLS can make a positive contribution to the cultivation of critical media competence among social media users. To this end, it will be analyzed whether and to what extent awareness, reflection, and empowerment are strengthened by the use of the artifact.

C. Trust

Niklas Luhmann [46] understands trust in the broadest sense as an elementary component of social life, interpreting it as a form of security, which can only be gained and maintained in the present. First and foremost, trust is needed to reduce a future of more or less undetermined complexity. According to Luhmann's understanding, the constant technical progress of society brings with it a simultaneous increase in complexity, which subsequently results in an increased need for trust. Thus, trust is a necessary condition to live and act with growing complexity in relation to modern events and dynamics [46]. However, trust is severely shaken by negative experiences [47], for instance caused by deception through disinformation. As MLS systems and algorithms become more complex, people increasingly regard them as ‘black boxes’ that defy comprehension in the sense that understanding an MLS’s decision requires growing amounts of specialized expertise and knowledge. Non-expert end-users are not able to retrace how the algorithmic code cascades led to a given decision [48]. Accordingly, there has been increased demand to offer the proper explanation for how and why a particular result was obtained [49]. Recent empirical evidence on algorithm acceptance [50] insinuates that explainability plays a heuristic role in algorithm and MLS service acceptance. Currently, however, research gives light to a controversy over whether the implementation of XAI-features actually helps increase user-trust or not. Shin [51] analyzed the impact of explainability in MLS on user trust and attitudes towards MLS and concluded that the inclusion of causability and explanatory features in MLS assists to increase trust as it helps users understand the decision-making process of MLS algorithms by providing transparency and accountability. In contrast, through their experiment on transparency and trust in MLS, [52] found that transparency features can actually affect trust negatively. These recent contradictory observations give rise to the need for further investigation of the effect of explainability on user trust. In the DeFaktS project, this research gap will be addressed through the evaluation of whether, and if so which, XAI elements increase user trust in the application.

III. Methodology

The goal of DeFaktS is to develop an artifact that is as close as possible to the needs of the subsequent user so that it contributes precisely to solving the above-mentioned issues. To implement this, the project is embedded in a design science research approach according to Peffers et al. [53], dividing the process into six steps: problem identification and motivation, definition of the objectives of a solution, design and development, demonstration, evaluation, and communication. Our research methodology for developing a taxonomy is based on the design science research paradigm, which seeks to address new knowledge about artificial objects that are designed to meet specific goals and benefit their users [54]. After identifying the problem and our motivation to develop a taxonomy of online disinformation, we defined the objectives of a solution; building an artifact that is intended to support researchers during the process of identifying and classifying (online) disinformation. Furthermore, the artifact serves as a basis for future design science research projects, the purpose of which is to investigate online disinformation and extend the given taxonomy [25]. This section gives insights into the design and development phase that researchers in the DeFaktS project are currently concerned with. Based on a structured literature review, we first build an artifact (TOD) for identifying and labeling disinformation. Then we evaluate the artifact by using it to create labels for a real-world dataset of factual news and ‘Fake News’. To this end, a group of experts evaluates the taxonomy by assessing its efficacy in developing labels for classifying social media content of interest in our specific domain. After the steps of demonstration and evaluation are completed, the artifact will be communicated via scholarly publications.

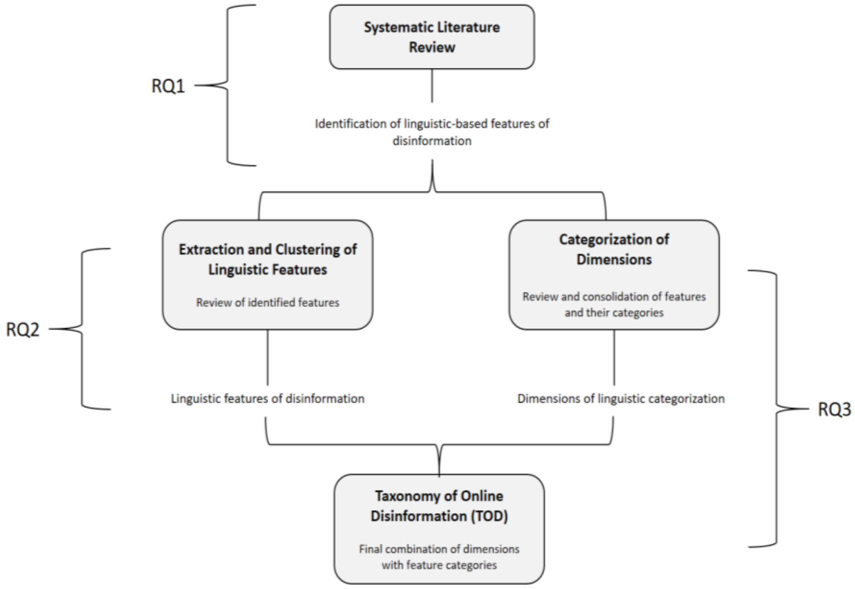

Systematic Literature Review
RQ1
Identification of linguistic-based features of disinformation
Extraction and Clustering of Linguistic Features
Review of identified features
Categorization of Dimensions
Review and consolidation of features and their categories
RQ2
Linguistic features of disinformation
Dimensions of linguistic categorization
RQ3
Taxonomy of Online Disinformation (TOD)
Final combination of dimensions with feature categories

Figure 1. Overall research outline.

Our approach, visualized in Figure 1, consists of two major parts that will be presented in the following. Initially, by conducting a systematic literature review [55], we gather all types of linguistic features of online disinformation in the literature. Subsequently, we cluster the empirical results in groups, supporting a linguistic-based 'Fake News' detection approach. Finally, we propose a novel, five-dimensional taxonomical framework, based on the categorization criteria found in the existing empirical literature. Our proceeding is guided by the following research questions:

1. What linguistic-based cues of (online) disinformation can be found in the empirical literature?
2. How can the linguistic features be clustered in an overarching schema?
3. How can the dimensions and categories resulting from the schema be conjugated into a taxonomy?

A. Systematic Literature Review

To comprehensively address our first research question, we conducted a systematic literature review based on Webster and Watson's [55] methodological guidelines. A thorough review contains pertinent literature on the subject and is not restricted to a particular research approach, collection of journals, or geographical area [55]. For this reason, we make use of large interdisciplinary databases to access all research fields relevant to our project. After carefully examining the literature on linguistic features and disinformation detection characteristics, we end up with an overview of descriptions that are frequently used to refer to different kinds and characteristics of disinformation content. However, the ad hoc definitions that each study introduces can cause conflicts or overlaps. Accordingly, the overall goal of our literature review is to make sense of the accumulated knowledge on categorizing disinformation, as well as to find patterns and identify key concepts in the literature in order to extend past research by synthesizing said knowledge into a useful taxonomy. For our review, we applied the following procedure:

1. Selection of our sources (digital libraries)
2. Definition of search terms
3. Application of each search term on selected sources
4. Selection of primary studies by use of inclusion and exclusion criteria on search results
5. Backward and forward search based on the selected primary studies

An automatic search was based on the following five primary sources of scientific databases to identify relevant publications: IEEE Xplore Digital Library, Scopus, Web of Science, Springer Link and Google Scholar.

We conducted several pilot searches based on our research topics to compile a preliminary list of papers. The search terms that best suited our research objectives were then defined using those as the foundation for the systematic review. The utilized search phrases restricted to abstract and title are listed in the following:

a. "fake news classification"
b. "disinformation classification"
c. "linguistic fake news detection"
d. "linguistic disinformation detection"
e. "linguistic fake news classification"
f. "linguistic disinformation classification"

For the next phase of our research, the following three inclusion and exclusion criteria were formulated:

1. We excluded sources that approached the issue of disinformation solely from a computational standpoint, proposing technical solutions based on, for instance, machine learning and statistical models to categorize news articles into predefined categories automatically, such as fake or real, as well as mere performance evaluations of such models.
2. Publications that mention specific categories or characteristics of false information without making an effort to classify them systematically or even to explain the proposed categories were excluded. This is used to describe sources where the disinformation phenomenon is either not a central concept (such as papers that happen to use terms like 'Fake News'), or they mention specific types of false information outside of a general framework or classification model and are therefore non-exhaustive or indicative.

   In the interest of common scientific understanding, only papers written in English were included.

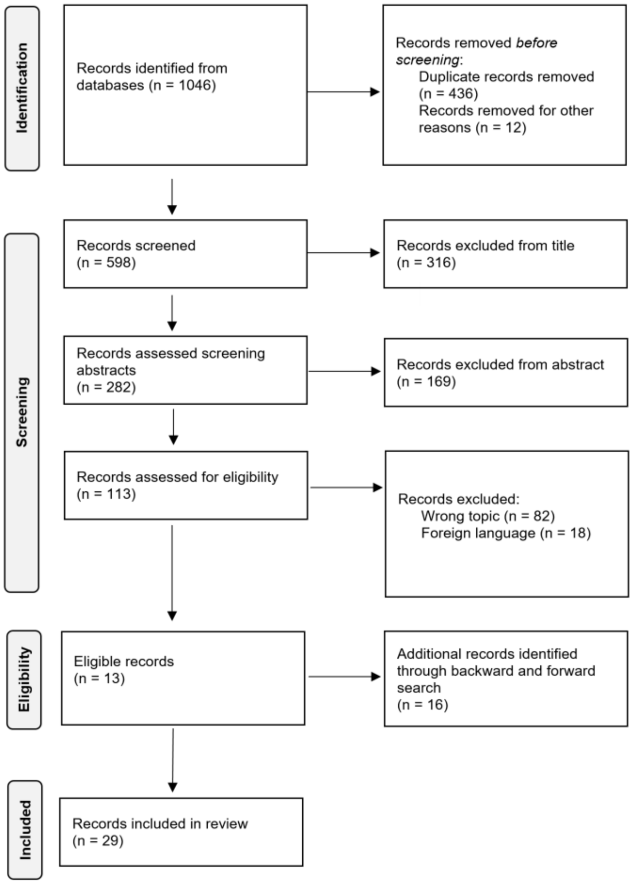

Identification
Records identified from databases (n = 1046)
Records removed *before screening*:
Duplicate records removed (n = 436)
Records removed for other reasons (n = 12)
Screening
Records screened (n = 598)
Records excluded from title (n = 316)
Records assessed screening abstracts (n = 282)
Records excluded from abstract (n = 169)
Records assessed for eligibility (n = 113)
Records excluded:
Wrong topic (n = 82)
Foreign language (n = 18)
Eligibility
Eligible records (n = 13)
Additional records identified through backward and forward search (n = 16)
Included
Records included in review (n = 29)

Figure 2. Prisma flow diagram.

Our search results, including the citations from the mentioned libraries, identify thirteen primary studies from six different disciplines (e.g., computer science, linguistics, psychology and media studies) where linguistic frameworks for disinformation detection are introduced. Figure 2 presents in detail the selection process of both records found through database searching and records identified through an additional backward and forward search based on the initial records, resulting in 29 papers included in our review. Our first goal was to identify linguistic-based cues of online disinformation in the empirical literature (RQ1). For addressing RQ2, we then extracted the identified features of disinformation and clustered them by similarities in an overarching schema in order to prepare our findings for RQ3.

B. Taxonomy of Online Disinformation

The overall goal of our research is to create a taxonomy of online disinformation, called TOD, that helps create a common understanding of what constitutes ‘Fake News’ or disinformation, provides a list of categories and detection characteristics and can be used to develop labels that can be applied to German using diverse ‘real world’ datasets (RQ3). After examining the findings from RQ1 and RQ2, we identify and extract the dimensions and categorization criteria resulting from our examination to select relevant and recurring ones. Our conclusive step is to systematically organize and map them into a taxonomy. In the context of our project, the final taxonomy shall support researchers to precisely label the datasets that will be used to train the DeFaktS AI. Beyond the scope of the DeFaktS project, offering a comprehensive and fine-grained taxonomy could also be utilized for educational purposes. Since online disinformation may influence people’s actions [56], considering the issue of classifying online disinformation from a global viewpoint could help to prevent having significant repercussions in real-life settings. Additionally, our research may also be useful in fields like artificial intelligence, where it is crucial to encode real-world concepts and entities consistently and methodically. A fact-checking or disinformation detection system will be able to produce the most accurate and understandable results the more clearly defined a particular type of disinformation is. The ‘Liar, Liar Pants on Fire’ dataset [57] and the ‘Fake News Corpus’ [58] are just two examples of the numerous ‘Fake News’ datasets that are currently available and used to research and develop detection models with utterly different labeling schemes. So far, the performance of computational models using various conceptual frameworks is not directly comparable, which makes it difficult to define the state of the art in science and industry and ultimately hinders the advancement of research. However, it is crucial for academics and professionals from various fields to come to consensus on this complicated subject, not only about the macro-level notions but also, if feasible, regarding the lower level of more specific attributes and subcategories.

As Figure 3 shows, the process starts with defining the meta-characteristic (1) based on the purpose of the taxonomy, that is linguistic cues of disinformation. Subsequently, the ending conditions are determined (2). In our case, we chose an empirical-to-conceptual approach, gathering empirical results through our systematic literature review from which we identify and extract common characteristics (3, 4). These characteristics are then grouped into dimensions to create and potentially revise the taxonomy (5). Practically, once a set of traits has been determined through the review, they can be formally categorized using statistical methods or arbitrarily by means of a manual or graphical process. The resulting groups define the taxonomy's initial dimensions. Since our method is iterative, conditions must be met in order to know when to stop (6).

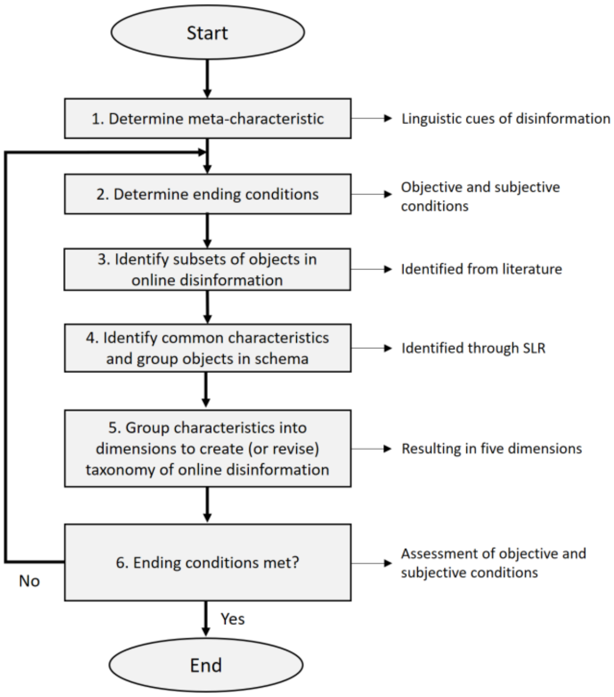
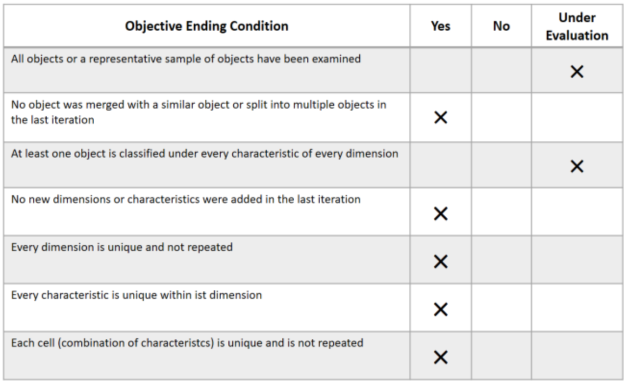

Start
1. Determine meta-characteristic
Linguistic cues of disinformation
2. Determine ending conditions
Objective and subjective conditions
3. Identify subsets of objects in online disinformation
Identified from literature
4. Identify common characteristics and group objects in schema
Identified through SLR
5. Group characteristics into dimensions to create (or revise) taxonomy of online disinformation
Resulting in five dimensions
6. Ending conditions met?
Assessment of objective and subjective conditions
No
Yes
End

Figure 3. Taxonomy development process [25].

There are both objective and subjective conditions [25]. Throughout the procedure, we check to see if the ending conditions have been satisfied with the taxonomy's current iteration at the conclusion of either of these steps (Tables 1 and 2). Conditions must be examined on both the objective and subjective level. If the objective conditions have been satisfied, it is necessary to investigate the subjective conditions. Both the objective and the subjective conditions must be satisfied for the method to be complete. These characteristics make up the prerequisites for a taxonomy's usefulness, but they do not always specify the sufficient requirements. Nevertheless, by crafting strong justifications for a taxonomy's utility, they can provide researchers with direction and serve as bases for descriptive evaluations based on reasoned argument. The method's output, or the taxonomy it produces after the design science building phase, needs to be assessed for usefulness. However, establishing the necessary conditions for usefulness is challenging, and ultimately, determining usefulness may depend on whether or not others find it useful [25].

Table 1 displays the current status regarding the objective ending conditions. While most of the requirements are satisfied, the assessment of the classification of a sample of objects is still ongoing and therefore marked as 'under evaluation'.

TABLE I. ASSESSMENT OF THE OBJECTIVE ENDING CONDITIONS FOR TOD.

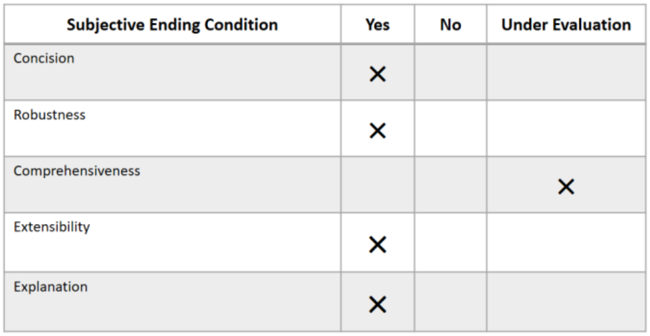


Objective Ending Condition	Yes	No	Under Evaluation
All objects or a representative sample of objects have been examined			×
No object was merged with a similar object or split into multiple objects in the last iteration	×		
At least one object is classified under every characteristic of every dimension			×
No new dimensions or characteristics were added in the last iteration	×		
Every dimension is unique and not repeated	×		
Every characteristic is unique within ist dimension	×		
Each cell (combination of characteristcs) is unique and is not repeated	×		

In addition to the objective ending conditions, [25] suggests that a useful taxonomy has the following subjective attributes:

1. It is concise: Because an extensive classification scheme with many dimensions and many characteristics may exceed the cognitive load of the researcher and be challenging to understand and apply, a taxonomy should only contain a small number of dimensions and a small number of characteristics in each dimension.
2. It is robust: A useful taxonomy should have sufficient dimensions and attributes to distinguish the objects of interest. A taxonomy with few dimensions and traits might not be able to distinguish between objects effectively.
3. It is comprehensive: There are two possible interpretations for this condition. One interpretation is that all known objects within the domain under consideration can be classified by a useful taxonomy (requirement of completeness). The second interpretation holds that all of an object's dimensions should be included in a taxonomy that is useful.
4. It is extendible: When new kinds of objects are discovered, a useful taxonomy should permit the inclusion of new dimensions and characteristics within an existing dimension. A taxonomy that cannot be expanded may quickly become outdated. In other words, it is dynamic rather than static.
5. It is explanatory: A useful taxonomy includes dimensions and traits that aid in our understanding of the objects by usefully elucidating their nature rather than exhaustively describing every aspect of the objects under study or the objects of the future.

TABLE II. ASSESSMENT OF THE SUBJECTIVE ENDING CONDITIONS FOR TOD.



Subjective Ending Condition	Yes	No	Under Evaluation
Concision	×		
Robustness	×		
Comprehensiveness			×
Extensibility	×		
Explanation	×		

Once again, Table 2 shows that most of the required subjective ending conditions are met, while the taxonomy's comprehensiveness is currently still under critical evaluation using real data. Furthermore, we reviewed the results of our systematic literature review considering the more granular level of their proposed features. We observed many commonalities but also differences at both the category and dimension levels. In order to make sense of the patterns and contradictions, we applied some general rules during the processing of the data.

a. Removal of types and definitions that are either generic (e.g., yellow press) or too technical (e.g., deep fakes).
b. Removal of duplicates and synonyms to avoid repetitions and overlaps.
c. Removal of types and definitions that were incorrectly categorized as disinformation (e.g., misinformation).

The fact that not all types of disinformation have the same degree of deceitfulness or harmful effects made the step of refining the disinformation taxonomy one of our biggest challenges, and some of them could not be categorized as disinformation. For instance, 'fabrication' is a more serious offense than 'hyperpartisanship' or 'clickbait', the latter of which has generated a lot of discussion. In our most recent iteration, this was addressed by adding a new dimension that deals with the degree of veracity. We completed our research goal, developing a taxonomy of online disinformation after taking the aforementioned information into account. As our goal is to create a useful taxonomy [59], our final test, then, is to examine the resulting taxonomy for its usefulness for the intended users and the intended purpose. The users of the TOD were projected to be researchers, journalists and developers of tools for disinformation detection, and their purpose was to distinguish among truthful and deceptive online content based on linguistic assessment. Nickerson et al. [25] claim that under some circumstances, such as possible collisions in the requirements for a taxonomy to be useful, conflicting criteria may need to be resolved by the researcher. This factor will be taken into account in a later research phase testing the usability of our framework.

IV. PRELIMINARY RESULTS

After our last iteration, we cannot identify any new characteristics and dimensions from the studies under review. Since in our case all ending conditions that can be met before putting the TOD into practice are satisfied, our final framework consists of five dimensions. The first dimension covers **different types of 'Fake News'**, splitting into subtypes (e.g., Trolling or Clickbait) and themes (e.g., pseudoscientific, commercial or political). Our second dimension contains **complexity features** that help to calculate the complexity and readability of the text, giving hints on its truthfulness. It serves users of the TOD to assess the informational content and textual structure of content under examination. A third dimension encompassing **psycho-linguistic features** describes attitudes, personas, behaviors, and emotions. This dimension, which includes the frequency of emotion words and informal language, helps to illustrate and quantify the cognitive process and individual concerns that underlie the writings. With a fourth dimension, we added **stylistic features** that shall reflect the style of the writers and syntax of the text, such as the number of verbs and nouns as well as the usage of certain terminologies. As mentioned before, disinformation content can differ strongly in its deceitfulness. For this reason, our fifth dimension accommodates **grades of veracity** ranging from 'No factual content' to 'Mostly true' to facilitate the evaluation of different kinds of 'Fake News' corresponding with our first dimension.

In the next steps, the current version of our taxonomy will be evaluated by a group of experts in the domain of research on online disinformation. Following on from this, it will be the subject of a workshop in which researchers will develop labels based on the TOD, which will then be used to label a 'Fake News' dataset that will form the basis for training the DeFaktS AI.

V. CONCLUSION AND FUTURE WORK

The research presented in this paper seeks to provide novel perspectives on the rapidly expanding field of combating online disinformation in a methodical and organized manner. Our goal was to discover and categorically define the many underlying linguistic features in the sphere of deceptive information, which was motivated by the lack of a conventionally accepted domain language. The concrete benefit of the developed TOD is, on the one hand, to make the phenomenon of disinformation as such more tangible, to achieve a common understanding of disinformation among researchers in the DeFaktS project, and to help answer the question of how disinformation in social media can be recognized as such. On the other hand, by unifying numerous study results on the linguistic detection of ‘Fake News’, our taxonomy offers researchers and actors in educational work a framework that provides a systematic overview of the scientific findings from the domain to date. By publishing the TOD and sharing it with a broad community at a later stage, we also hope to contribute to simplifying and standardizing the labeling of data for ‘Fake News’ detection, and thereby making it more transparent.

As we approached this intricate and vast field, we faced some substantial challenges. An issue we encountered was the large amount of research output produced by the latest wave of Big Data, AI, and MLS tools. Despite the abundance of scientific studies in the area, we discovered that the majority of them introduce singular and ad hoc solutions, leading to a fragmentation issue. The main objective of this type of research is still to suggest effective and precise algorithmic approaches as well as to evaluate their performance, so in most cases the justification and conceptual model are not sufficiently explained. In addition, we discovered that depending on its nature, disinformation can vary greatly in its veracity, which may cause difficulties in classification by means of a schema. To resolve this concern, we added a dimension to the TOD called ‘grades of veracity’, allowing us to address the various subtypes and topics that fall under the definition of disinformation. Yet, we anticipate the emergence of new types of disinformation and their associated characteristics given the dynamic nature of the domain, potentially causing the need for a revision of our taxonomy and the dimensions it entails. Because of this, we invite researchers to evaluate and validate the framework in the future to identify potential new dimensions or categories that may alter or extend our work.

We also concluded that multidisciplinary approaches are essential for comprehending and developing strategies and tools to combat the spread of deceptive information. Despite the field's close ties to political communication theory, we think that modern disinformation demonstrates traits that necessitate the use of additional analytical tools. Digital communities that exhibit distinctive traits that are difficult to compare to the past are where disinformation is flourishing [56]. Furthermore, disinformation also encompasses forms outside of the political sphere, such as fake reviews and pseudoscience. Finally, the development of (semi-)automated fact-checking tools is predicted by the recent impressive advancements in technologies like machine learning. These currently observable dynamics call for more interdisciplinary research on the domain that we would like to encourage with our contribution.

References

[1] I. Bezzaoui, J. Fegert, and C. Weinhardt, “Distinguishing Between Truth and Fake: Using Explainable AI to Understand and Combat Online Disinformation”, The 16th International Conference on Digital Society, 2022.

[2] K. Shu, A. Sliva, S. Wang, J. Tang, and H. Liu, “Fake News Detection on Social Media: A Data Mining Perspective”, ACM SIGKDD Explorations Newsletter, 19, 2017.

[3] K. Shu, A. Bhattacharjee, F. Alatawi, T.H. Nazer, K. Ding, M. Karami, and H. Liu, “Combating Disinformation in a Social Media Age”, WIREs Data Mining and Knowledge Discovery, 10, 1–23, 2020.

[4] D. McQuail, “Media performance: Mass communication and the public interest”, Thousand Oaks, CA: Sage. M. Young,The Technical Writer’s Handbook, Mill Valley, CA: University Science, 1992.

[5] J. Strömbäck, “In search of a standard: Four models of democracy and their normative implications for journalism”, Journalism Studies, (6:3), pp. 331-345, 2005.

[6] J. Groshek and K. Koc-Michalska, “Helping populism win? Social media use, filter bubbles, and support for populist presidential candidates in the 2016 US election campaign”, Information Communication and Society, (20:9), pp. 1389-1407, 2017.

[7] H. Holone, “The filter bubble and its effect on online personal health information”, Croatian Medical Journal, (57:3), pp. 298–301, 2016.

[8] K. Sharma, Y. Zhang, and Y. Liu, “COVID-19 vaccines: characterizing misinformation campaigns and vaccine hesitancy on twitter”, Retrieved May 2022. arXiv preprint arXiv:2106.08423, 2021.

[9] E.C. Tandoc Jr, “The facts of fake news: A research review”, Sociology Compass, 13(9), e12724, pp. 1-9, 2019.

[10] L. Munn, “Angry by design: toxic communication and technical architectures”, Humanities and Social Sciences Communications, 7(1), pp. 1-11, 2020.

[11] K. Nelson-Field, E. Riebe, and K. Newstead, “The emotions that drive viral video”, Australasian Marketing Journal, 21(4), pp. 205–211, 2013.

[12] K.A. Rosińska, “Disinformation in Poland: Thematic classification based on content analysis of fake news from 2019”, Cyberpsychology: Journal of Psychosocial Research on Cyberspace, 15(4), 2021.

[13] H.Q. Abonizio, J.I. de Morais, G.M. Tavares, and S. Barbon Junior, “Language-independent fake news detection: English, Portuguese, and Spanish mutual features”, Future Internet, 12(5), 87, 2020.

[14] B. Horne and S. Adali, “This just in: Fake news packs a lot in title, uses simpler, repetitive content in text body, more similar to satire than real news”, 11(1), pp. 759–766, 2017.

[15] D. Schreiber, C. Picus, D. Fischinger, and M. Boyer, “The defalsif-AI project: Protecting critical infrastructures against disinformation and fake news/Das Projekt defalsif-AI: Schutz kritischer Infrastrukturen vor Desinformation und Fake News”, Elektrotechnik und Informationstechnik, Vol. 138 (7), pp. 480–484, 2021.

[16] X. Zhou and R. Zafarani, “A survey of fake news: Fundamental theories, detection methods, and opportunities”, ACM Computing Surveys (CSUR), 53(5), pp. 1-40, 2021.

[17] D. Rohera et al., “A Taxonomy of Fake News Classification Techniques: Survey and Implementation Aspects”, IEEE ACCESS, 10, 2022.

[18] W. Shahid et al., “Detecting and Mitigating the Dissemination of Fake News: Challenges and Future Research Opportunities, IEEE Transactions on Computational Social Systems, 2022.

[19] W. Ansar and S. Goswami, “Combating the menace: A survey on characterization and detection of fake news from a data science perspective”, International Journal of Information Management Data Insights, 1(2), 2021.

[20] F. I. Adiba, T. Islam, M.S. Kaiser, M. Mahmud, and M.A. Rahman, “Effect of corpora on classification of fake news using naive Bayes classifier”, International Journal of Automation, Artificial Intelligence and Machine Learning, 1(1), pp. 80–92, 2020.

[21] B. Akinyemi, O. Adewusi, and A. Oyebade, “An Improved Classification Model for Fake News Detection in Social Media”, International Journal of Information Technology and Computer Science (IJITCS), 12(1), pp. 34–43, 2020.

[22] M. Fayaz, A. Khan,M. Bilal, and S.U. Khan, “Machine learning for fake news classification with optimal feature selection”, Soft Computing, pp. 1–9, 2022.

[23] Y. Lasotte, E. Garba, Y. Malgwi, and M. Buhari, “An Ensemble Machine Learning Approach for Fake News Detection and Classification Using a Soft Voting Classifier”, European Journal of Electrical Engineering and Computer Science, 6(2), pp. 1–7, 2022.

[24] R.A. Hirschheim, H.K. Klein, and K. Lyytinen, “Information Systems Development and Data Modeling: Conceptual and Philosophical Foundations”, Cambridge University Press, Cambridge, 1995.

[25] R.C. Nickerson, U. Varshney, and J. Muntermann, “A Method for Taxonomy Development and its Application in Information Systems”, European Journal of Information Systems, 22, pp. 336–359, 2013.

[26] R.L. Glass and I. Vessey, “Contemporary application-domain taxonomies”, IEEE Software 12(4), pp. 63–76, 1995.

[27] J. Iivari, “A paradigmatic analysis of information systems as a design science”, Scandinavian Journal of Information Systems 19(2), pp. 39–64, 2007.

[28] J. Delcker, Z. Wanat, and M. Scott, „The coronavirus fake news pandemic sweeping WhatsApp”, Politico, Retrieved May 2022 from https://www.politico.eu/article/the-coronavirus-covid19-fake-news-pandemic-sweeping-whatsapp- misinformation/, 2020.

[29] S. Yu and D. Lo, “Disinformation detection using passive aggressive algorithms”, ACM Southeast Conference, Session 4, p. 324f, 2020.

[30] P. K. Verma, P. Agrawal, I. Amorim, and R. Prodan, “WELFake: Word embedding over linguistic features for fake news detection”, IEEE Transactions on Computational Social Systems, 8(4), pp. 881–893, 2021.

[31] M. Mahyoob, J. Al-Garaady, and M. Alrahaili, “Linguistic-based detection of fake news in social media.” Forthcoming, International Journal of English Linguistics, 11(1), pp. 99-109, 2020.

[32] H. Alsaidi and W. Etaiwi, “Empirical evaluation of machine learning classification algorithms for detecting COVID-19 fake news”, Int. J. Advance Soft Compu. Appl, 14(1), pp. 49-59, 2022.

[33] W. H. Bangyal et al., “Detection of Fake News Text Classification on COVID-19 Using Deep Learning Approaches”, Computational and Mathematical Methods in Medicine, pp. 1-13, 2021.

[34] L. Bozarth and C. Budak, “Toward a better performance evaluation framework for fake news classification”, Proceedings of the international AAAI conference on web and social media, 14, pp. 60–71, 2020.

[35] C. Lai et al., “Fake news classification based on content level features”, Applied Sciences, 12(3), p. 1116, 2022.

[36] A. M. Guess et al., “A digital media literacy intervention increases discernment between mainsitream and false news in the United States and India”, PNAS, 117(27), pp. 15536–15545, 2020.

[37] G. Pennycook and D. G. Rand, “Lazy, not biased: Suceptibility to partisan news is better explained by lack of reasinong than by motivated reasoning”, Cognition, pp. 1–12, 2018.

[38] D. Ribes Lemay et al., “Trust indicators and explainable AI: A study on user perceptions”, IFIP Conference on Human-Computer Interaction - INTERACT 2021, pp. 662–671, 2021.

[39] K. Siau and W. Wang, “Building trust in artificial intelligence, machine learning, and robotics”, Cutter Business Journal, 31(2), pp. 47-53, 2018.

[40] P. Dahlgren, “Media and political engagement: Citizens, communication, and democracy”, Cambridge: Cambridge University Press, 2009.

[41] S. M. Jang and J. K. Kim, “Third person effects of fake news: Fake news regulation and media literacy interventions”, Computers in Human Behavior, 80, pp. 295–302, 2018.

[42] D. Rieger et al., “Propaganda und Alternativen im Internet - Medienpädagogische Implikationen. Propaganda and Alternatives on the Internet - Media Pedagogical Implications“, merz / medien + erziehung, (3), pp. 27-35, 2017.

[43] D. Kellner and J. Share ”Critical media literacy, democracy, and the reconstruction of education”, In D. Macedo & S. R. Steinberg, Media Literacy: A Reader, Peter Lang Publishing, pp. 3-23, 2007.

[44] D. Kellner and J. Share, “Toward critical media literacy: Core concepts, debates, organizations, and policy”, Discourse: studies in the cultural politics of education, 26(3), pp. 369–386, 2005.

[45] J. B. Schmitt, D. Rieger, J. Ernst, H. J. Roth, „Critical media literacy and Islamist online propaganda: The feasibility, applicability and impact of three learning arrangements”, International Journal of Conflict and Violence, 12, pp. 1–19, 2018.

[46] N. Luhmann, „Vertrauen“, Trust (5), UVK, 2014.

[47] F. Schwerter and F. Zimmermann, „Determinants of trust: The role of personal experiences”, Games and Economic Behavior, 122, pp. 413-425, 2020.

[48] D. Castelvecchi, “Can we open the black box of AI?”, Nature, 538, pp. 20–23, 2016.

[49] M. Ter Hoeve et al., „Do news consumers want explanations for personalized news rankings?” FARTEC, pp. 1–6, 2017.

[50] D. Shin, B. Zhong, F. A. Biocca, “Beyond user experience: What constitutes algorithmic experiences?” International Journal of Information Management, 52, pp. 1–11, 2020.

[51] D. Shin, “The effects of explainability and causability on perception, trust and acceptance: Implications for explainable AI”, International Journal of Human-Computer Studies, 146, pp. 1-11, 2021.

[52] T. Schmidt, F. Biessmann, T. Teubner, „Transparency and trust in artificial intelligence systems”, Journal of Decision Systems, 29(4), pp. 260–278, 2020.

[53] K. Peffers, T. Tuunanen, M. A. Rothenbergre, S. Chatterjee, “A design science research methodology for information systems research”, Journal of Management Information Systems, 24(3), pp. 45–77, 2007.

[54] H.A. Simon, “The Sciences of the Artificial”, The MIT Press, Cambridge, MA, 1969.

[55] J. Webster and R.T. Watson, “Analyzing the Past to Prepare for the Future: Wiriting a Literature Review” MIS Quarterly, 26(2), pp. 13–23, 2002.

[56] E. Kapantai, A. Christopoulou, C. Berberidis, and V. Peristeras, “A Systematic Literature Review on

Disinformation: Toward a Unified Taxonomical Framework”, New Media & Society, 23(5), pp. 1301–1326, 2021.

[57] W.Y. Wang, “liar, liar pants on fire: A new benchmark dataset for fake news detection”, arXiv preprint arXiv:1705.00648, 2017.

[58] M. Szpakowski, “Fake News Corpus Dataset”, https://github.com/several27/FakeNewsCorpus, 2020.

[59] A.R. Hevner, S.T. March, J. Park, and S. Ram, “Design science in information systems research”, MIS Quarterly 28(1), pp. 75–105, 2004.

Detecting Manipulated Wine Ratings with Autoencoders and Supervised Machine Learning Techniques

Michaela Baumann
Business Intelligence / Analytics Competence Center
NÜRNBERGER Versicherung
Nürnberg, Germany
email: michaela.baumann@nuernberger.de
ORCID: 0000-0001-5066-9624

Michael Heinrich Baumann
Department of Mathematics
University of Bayreuth
Bayreuth, Germany
email: michael.baumann@uni-bayreuth.de
ORCID: 0000-0003-2840-7286

***Abstract*—In this study, we analyze the ability of different machine learning methods to detect manipulated wine ratings. We consider autoencoders, regression models (neural networks, support vector machines, random forests) and classification models (support vector machines, random forests) and two different kinds of manipulation strategies. We find that autoencoders perform best on unmanipulated test data, i.e., their reconstruction error is smaller than the supervised models' prediction error. However, on the manipulated test data, the supervised models outperform autoencoders. This is interesting since autoencoders are generally used for outlier detection. When comparing only the supervised methods, we find that, basically, both support vector machines and random forests perform and detect better than regression neural networks. Additionally, the optimization and training times for these two model types are smaller. In order to consider a relatively large grid of hyperparameters especially for the neural networks, we introduce a hyperparameter tuning method called sequential accumulative selection. To sum up, when trying to detect manipulations, different methods have usually both advantages and disadvantages.**

Keywords—anomaly detection; manipulation identification; wine preferences; artificial neural networks; autoencoders; support vector machines; random forests.

I. Introduction

The work at hand is an extended version of [1]. Some parts are identical to the conference paper [1] (May 22 version and Oct. 22 version), however, we especially modified and extended the methodology (Section IV) by considering a larger set of models and different data manipulation strategies and, therefore, get new and more detailed results (Section VI).

In a world of increasingly differentiated products and customers who frequently change their buying behavior, it is difficult to assess whether the price-performance ratio is appropriate before making a purchase. An important and much-used assistance in such buying decisions are ratings. In this study, we are going to approach the question of whether and how manipulated ratings can be detected using wine quality ratings as an example. When ratings come from an official or non-official authority (such as Gambero Rosso's *Vini d'Italia* [2], Robert Parker's *The Wine Advocate* [3], *Gault&Millau* [4], or *Guide Michelin* [5], when dealing with wines, hotels, restaurants, or related topics), it is possible to verify with little effort whether ratings given by a merchant or producer are genuine by simply looking up the relevant work. However, since by far not all wines are represented and rated in one of the works published by an authority, there are countless other ratings. These other ratings, which are not given by an authority, are difficult to verify for authenticity, and it might even be possible that they are not objective, but rather paid for by someone. In the following, we are going to show possibilities for detecting such manipulated or faked ratings.

A very basic idea for how to identify manipulated ratings would be via (linear) regressions (cf. [6]). That means, when we have other, exactly measurable features, such as alcohol content, pH value, or density, we can learn how to predict the rating using these independent variables on correctly rated data objects. Ratings that differ (strongly) from the predicted ones on unseen data might be suspicious. However, a linear regression does not lead to good results in our case, i.e., when trying to detect manipulated wine ratings. Thus, the research question is how manipulated wine ratings may be detected in a better way.

Since artificial neural networks are currently en vogue, one can of course use a regression by means of a neural network (cf. [7]–[10]). Note that a linear regression is the same as an exactly trained, fully connected neural network without any hidden layer with linear activation functions (i.e., id resp. pass-through), when adding a dummy column (filled with 1s) in the data for the intercept and using *Mean Squared Error* (MSE) as loss. Regressions based on shallow or deep neural networks are likely to outperform a linear regression. Other machine learning techniques for performing regression tasks are, for example, support vector machines [11][12] or random forests [13][14].

Especially when dealing with outlier detection, so-called autoencoders (resp. reconstruction networks or autoassociative neural networks) are a common means [15]–[18]. Autoencoders consist of two parts (i.e., two regressions), an encoder and a decoder. The encoder compresses the input data to a lower dimensional representation usually referred to as the code; the decoder takes the code as input and aims to reconstruct the original input.

Given a well trained autoencoder, when the input and the output differ (strongly), the data might be manipulated (or in other contexts: an outlier, an anomaly, fraudulent, or suspicious). Note that there are much more application areas of autoencoders, such as dimensionality reduction, data compression, or denoising. Although it is in principal assumed that the quality depends on the other features, the autoencoder does not use this dependence information, that is, the quality and all other features are considered as coequal input (and output) variables. Since the autoencoder does not use all the information that is actually available, it would be very interesting if it nevertheless achieved better results.

In the work at hand, we investigate how *Regression Neural Networks* (RNNs), *Support Vector Machines* (SVMs), *Random Forests* (RFs) and *Neural Network based Autoencoders* (NNAs) can be used to identify manipulated data. Additionally, as benchmark models we use a linear regression (*Linear Model;* LM; see [6]) and an autoencoder that implements two linear regressions (*Benchmark Autoencoder;* BA; see Section IV-D).

In the work at hand, we summarize RNN, SVM, RF and LM under the term *supervised methods,* since these models are trained with an explicit label, the wine quality. The autoencoders NNA and BA are not referred to as supervised methods. Usually, autoencoders fall into the category of unsupervised methods since there is no explicit target feature or label, but only covariates. This reflects the usage of autoencoders when the code layer is of interest, e.g., when conducting dimensionality reduction or data compression. Even though the training process does not differ, when the reconstructed input explicitly is of interest, autoencoders may in this case be called *autoassociative* or *self-supervised methods* [19][20].

There clearly are several other data analytics methods that might be applied, especially since the number of those techniques keeps growing (see [21][22]), however, the methods used here are conceptually different from each other and therefore cover a reasonable range of possible methods.

We find that both regression and classification techniques are suitable for detecting manipulated wine ratings. Autoencoders as means for uncovering outliers in data sets do not detect the manipulated data objects as well as the supervised methods. Although we provide a relatively large grid of possible hyperparameters for the neural network based models, both RFs and SVMs show a better detection performance than the RNN and the NNA. Further, the variability in the results of the neural network based models is quite high.

The remainder of this paper is organized as follows: Section II reviews both the literature on wine data analysis and those on anomaly detection in general while Section III specifies the data we are using. Section IV depicts our methodology step by step and Section V describes the preparatory activity for the hyperparameter optimization of the neural networks called sequential accumulative selection. Section VI presents the results. Finally, Sections VII and VIII conclude and describe possibilities for ongoing work.

II. Literature Review

The closely related literature roughly splits into two groups, namely data analytics of wine quality and general outlier/anomaly detection resp. fraud identification. The analytics of wine quality mostly covers the prediction of wine ratings based on measurable features. Cortez et al. [23][24] compare several data mining regression methods for predicting wine preferences based on easily available data during the certification of wines. In this context, they originally published the two datasets that are also used in the work at hand. They use the vinho verde white wine dataset [24] and both the vinho verde red wine and white wine datasets [23]. The data mining methods for predicting wine quality in both papers are neural networks, i.e., multilayer perceptrons, and SVMs. Besides these papers, also the importance of the selection of the most relevant features before predicting the wine quality with machine learning regression methods is investigated for the vinho verde datasets [25]. Here, a linear regression is used for assessing the importance of the variables. Neural networks and SVMs are then trained for the actual prediction of the wine quality via regressions. The neural network and SVM results are compared when using all available variables or only the most important ones as assessed by the linear regression. The vinho verde white wine dataset is used for classifying wine preferences via fuzzy inductive reasoning [26]. To increase statistical confidence, in [26], the evaluation process is repeated 20 times. Deep neural regression networks are applied for predicting wine ratings on the vinho verde datasets [27]. In [27], the authors train the neural networks separately for the red and white wine datasets and find that different network architectures are needed for the two datasets. These results are then compared to a multiclass SVM, where the neural networks outperform the SVM.

An expert model consisting of several submodels for different types of input variables for assessing wine quality is developed to assist the winemakers in their business [28]. This model is evaluated on 45 data samples from southern France consisting of altogether 137 variables (vineyard variables and enological variables). In other works, the effect of weather and climate changes as well as the effect of expert ratings on the prices of Bordeaux wines are analyzed [29][30]. With this, the efficiency of the Bordeaux wine market is assessed. Tree models are used for predicting the relative quality of German Rhinegau Riesling considering terrain characteristics obtained through cartographic studies [31]. Due to unbalances in the original seven target categories, the Riesling quality classification of [31] is developed for a target variable mapped from the original seven to three more equally balanced categories. In another work, a framework is developed that automatically finds an appropriate set of classifiers and hyperparameters via evolutionary optimization for predicting wine quality for arbitrary wine datasets [32]. That work also gives a good overview over further research concerning the prediction of wine quality with several machine learning methods, such as k-nearest neighbors [33][34], naive Bayes [35][36], or RFs [34][36].

Having in mind the literature briefly reviewed above, which predicts wine ratings or conducts data analyses of wine quality, the work at hand contributes by connecting wine rating predictions and anomaly detection. The topic of outlier/anomaly detection and fraud identification is addressed in a lot of related work in various contexts (see, for example, the surveys and summaries [37]–[41]) and we can only touch on this broad topic here. Generally, according to Chandola et al., "*Anomaly detection* refers to the problem of finding patterns in data that do not conform to expected behavior" [37]. Usually, anomalies have to be identified throughout the analysis of data so that they can be treated separately and do not distort the results of the analysis of "normal" data. However, in the case of fraud and also in our case of manipulation detection they are of special interest (cf. [37]). Fraudulent and manipulated data objects inhibit abnormal patterns but they try to appear as normal. The detection of anomalies, especially of intentional, malicious anomalies, such as fraud or manipulation, is very challenging and there are many approaches that try to accomplish this task. The approaches basically fall in one of the following three categories [38]:

- Unsupervised methods (e.g., clustering); labels are not needed here and new patterns (normal ones and outliers) may be processed correctly.
- Supervised methods (e.g., classification); these need prelabeled data, however, anomalies are usually very rare and the labeled datasets are, thus, highly unbalanced; new patterns are unlikely to be processed correctly.
- Semi-supervised methods; normal behavior is known, i.e., (a part of) the training data is labeled as normal, and new, unlabeled data objects are compared to the normal case.

When we apply the models mentioned in Section I for detecting manipulated wine ratings, we use them in a semi-supervised fashion. This means, the supervised models RNN, SVM, and RF are trained for predicting the target feature "wine quality" and the NNA is trained for reconstructing all features (autoassociative resp. self-supervised). In each case, this training is carried out on unmanipulated wine data. Whether the wine data is manipulated or not is a second label/target feature that is not present in the training process or, in other words, that is the same (namely: not manipulated) for the whole training data set. For assessing the manipulation detection ability of the models, the supervised models' predictions and the autoencoder's reconstructed features of the test data, which reflect the normal, i.e., unmanipulated case, are compared to the unlabeled (in terms of manipulation) test data. The models themselves cannot be semi-supervised, but the detection approach as a whole is semi-supervised.

In addition to methods that require tabular data (a priori tabular data, but also image, audio, or video data transferred to tabular data) there are methods that operate on graph based data [42], which are especially useful when identifying anomalies in highly connected data. The approach of the work at hand falls into the third category, i.e., semi-supervised methods, and works on tabular data.

III. Data

The approach described in this work is applicable to various working areas (see Section VIII). We demonstrate it using wine data as an example because of the following reasons.

A rather simple advantage is the good data availability and (if no wine names or winemaker names are used) the innocuousness of the data. Further, the explaining variables (except for wine or winemaker names) are metric, clearly defined, and exactly measurable (e.g., alcohol content, acid, pH value, red/white). The used datasets further have a unique target feature and not a list of ratings (see also Section VIII).

We use the "Wine Quality Datasets" [23] from the Universidade do Minho [43], more specifically the datasets "White Wine Quality—Simple and clean practice dataset for regression or classification modelling" [44] and "Red Wine Quality—Simple and clean practice dataset for regression or classification modelling" [45] downloaded from *kaggle,* which are licensed under "Database Contents License (DbCL) v1.0," *Database: Open Database, Contents: Database Contents* [46]. Both datasets contain anonymized *vinho verde* wines and have the same twelve columns: eleven independent variables summarized in Figure 1 through boxplots and the dependent variable summarized in Figure 2 by means of a histogram. The dependent variable *quality* is the wine rating, which is supposed to depend on the other, explaining features, called independent. All values except for the ratings are in some meaningful physical unit, while the ratings range from 0 (very bad) to 10 (excellent) in integer steps [23]. The red wine dataset consists of 1,599 entries while the white wine data has 4,898 rows, leading to a combined data set with 6,497 rows and 13 columns.

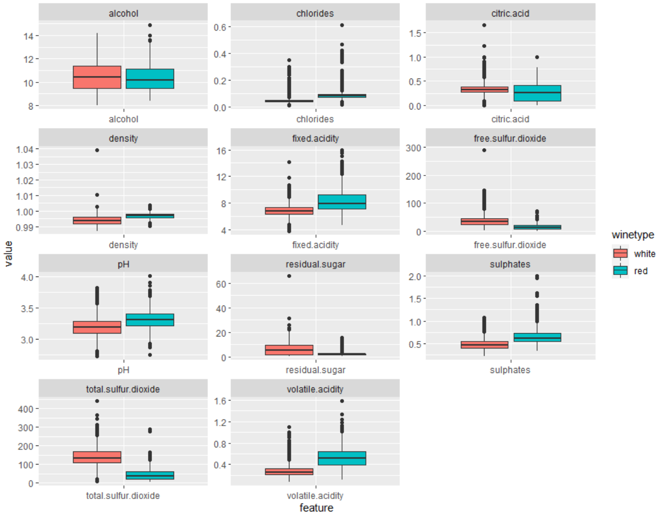


Figure 1. Summary of the distribution of the independent features for the white and red wines via boxplots

For the distinction of red and white wines we added a binary encoded categorical column to the union of both datasets. There already exist extensive analyses of the vinho verde datasets covering, among others, correlations, clusterings, distribution estimations, etc. Such statistics and many more analyses can be found in the work of Cortez et al. [23][24], in other papers [25]–[27], and further tutorials or notebooks [47]–[51].

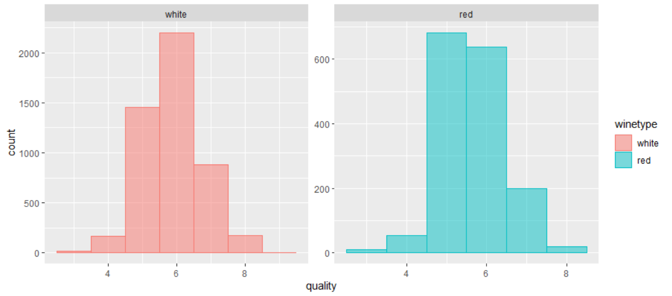


Figure 2. Summary of the distribution of the dependent variable for the white and red wines via histograms

IV. Methodology

As outlined in Section I, the aim of this work is to identify manipulated ratings. For this, we train several network based models, SVMs, and RFs on provided, correct data. We then make predictions on unseen data objects where we manipulate a certain part of these objects. We use two different kinds of manipulation strategies. In the first one, we increase the original rating of very low rated wines as this seems to be a “reasonable” manipulation in the context of wine ratings, e.g., when someone wants to increase sales numbers. In the second one, we take a random part of the test data and replace the wine rating by a value drawn randomly from the empirical distribution of the remaining, unmodified part of the test data. By comparing the potentially manipulated data and the predicted data we aim at identifying the manipulated data objects. We assume that objects for which the predicted values strongly differ from the provided data are more likely to be manipulated. We assess the models’ detection performance, i.e., their ability to identify manipulations through calculating the true and false positive rates when marking the most deviating data objects as suspicious. To prevent overfitting and account for other random effects we apply bootstrapping. That is, we repeat the process of randomly splitting the data and training the models such as, for example, also done in [26]. Finally, we take among others the median over the particular results.

In the following, we explain our methodology in detail. The implementation is done in `R` using the `Keras` library, which is an API to `TensorFlow`, for the neural networks, the `e1071` library for the SVMs, and the `caret` library with the `rf` option for the RFs.

A. Bootstrapping and Data Splitting

The bootstrapping is in our case a Monte Carlo like approach of repeatedly and independently splitting the complete dataset $\underline{all}$ (6,497 rows, 13 columns) with a ratio of 70:30 into training data (4,547 rows, 13 columns) and test data (1950 rows, 13 columns) 100 times: $\underline{all} = \underline{train} \,\dot{\cup}\, \underline{test}$. To make this process reproducible, we set an initial seed and randomly draw 100 seeds $(seed_1, \ldots, seed_{100})$. Before every splitting we explicitly set the seed to the respective run’s seed. Please note that when conducting the hyperparameter optimization for the different methods, the training data set $\underline{train}$ is split automatically inside the respective `R` functions into a development and validation data set or into the different folds when using a k-fold cross validation.

B. Data Manipulation

We use and compare two wine rating manipulation strategies. In the first one, we manipulate the 10% worst ranked test data by averaging the original rating and the highest possible rating (10) and rounding up. That is, we split the test data $\underline{test} = \underline{low} \,\dot{\cup}\, \underline{high}$ with a ratio of 10:90 (with a random tie breaking), manipulate $\underline{low} \mapsto \underline{low_{manip\_worst}}$ and get the manipulated test data

$$\underline{manip_{worst}} := \underline{low_{manip\_worst}} \,\dot{\cup}\, \underline{high}.$$

We also add a flag column to the manipulated test data for marking the manipulated entries for evaluation purposes.

In the second manipulation strategy, we randomly take 10% of the test data and replace the true rating by a rating value drawn from the empirical distribution of the remaining, unmodified part of the test data. That is, we split the test data $\underline{test} = \underline{random} \,\dot{\cup}\, \underline{rest}$ with a ratio of 10:90 (again with a random tie breaking). The manipulation replaces $\underline{random} \mapsto \underline{random_{manip\_random}}$ such that we get the manipulated test data

$$\underline{manip_{random}} := \underline{random_{manip\_random}} \,\dot{\cup}\, \underline{rest}.$$

Of course, in the second manipulation strategy it may happen for some or even for all wines, that the true rating and the manipulated rating are the same due to the random drawing of manipulated ratings. That is, we actually have a rating manipulation of at most 10% randomly drawn wines. Further, the changes of the true and the manipulated ratings may be rather small, especially compared to the changes induced by the first manipulation strategy. This is why we expect all detection methods to “perform,” i.e., to detect better on the test set manipulated with strategy one, i.e., on $\underline{manip_{worst}}$, than with strategy two, i.e., on $\underline{manip_{random}}$.

C. Data Normalization

As can be seen in Figure 1, the scales of the independent wine features differ strongly. Most of the machine learning models have problems with differing scales, which is why some data preprocessing, in our case the normalization of the data, is necessary. We normalize the independent features by min-max-scaling where $\underline{train}$ serves as reference. That means, also the test datasets are normalized with the minimum and maximum values of $\underline{train}$. For LM, RNN, SVM, and RF the target variable “quality” is not normalized. For BA and NNA, “quality” is an input variable like the others and, hence, normalized. To obtain comparable results, the performance of the regression models is normalized afterwards (using $\underline{train}$).

D. Models

We consider six different kinds of models: LM, RNN, SVM, RF, BA, and NNA. Except for the two autoencoders (BA and NNA), the other model types also appear in the related work concerning the prediction of wine quality, see Section II. The two simple models LM and BA are solely for benchmarking the general performance of the other models on the unmanipulated test data $\underline{test}$.

Because SVMs and RFs are made primarily for classification, we optimize and train SVM and RF not only as regression models, but also as classification models. This means, for regression, we treat the wine quality as a numeric variable with values in $[0, 10]$, for classification we treat it as a factor variable with values in $\{0, 1, \ldots, 10\}$. We refer to the classification models as SVMclass and RFclass, the regression models are further named SVM and RF.

Thus, we measure the manipulation detection performance for RNN, SVM, SVMclass, RF, RFclass, and NNA. The particular model configurations are described in Section IV-E. For the two simple models, the following holds: LM uses `R`'s `lm` function. BA is a fully connected, three layer network with input layer (size 14), code layer (size 4), and output layer (size 14). The input is the 13 dimensional data plus a constant column of 1s (intercept) in order to mimic two nested linear regressions. This is also the reason why linear activation functions and MSE as validation metric are used.

E. Hyperparameter Tuning

In every step during the bootstrapping, all models except the two simple ones are trained with hyperparameter optimization over a grid. Especially for the neural network based models RNN and NNA, the respective grids are quite large (see also, e.g., [27]). We apply the so-called sequential accumulative selection, see Section V, in a preceding step in order to possibly reduce the grid size for the actual tuning and training. The tuning of the neural network based models uses simple splits into development and validation data due to runtime issues. That means, we do not apply a k-fold cross validation here (at least not for $k > 1$). The hyperparameter grid after applying the sequential accumulative selection for RNN is:

- Activation function (hidden layers): `linear`, `softplus`, `ReLU`
- Activation function (output layer): `linear`
- Number of hidden layers: 1, 3, 5, 7
- Dropout rate: 0%, 5%, 10%
- Number of neurons in each hidden layer: 32, 64, 128
- Number of neurons the input layer: 12
- Number of neurons the output layer: 1
- Batch size: 32, 64
- Learning rate: 5%, 10%
- Patience for early stopping: 15
- Patience for learning rate reduction: 7
- Loss function: MSE
- Evaluation measure: *Mean Absolute Error* (MAE)
- Optimizer: `Adam`
- Number of epochs: 75
- Batch normalization: between every layer

The grid for NNA after the sequential accumulative selection is defined as follows:

- Activation function (hidden layers, except the code): `softplus`, `ReLU`
- Activation function (code layer and output layer): `linear`
- Number of hidden layers (except code layer): 4, 6
- Dropout rate: 0%
- Number of neurons in each hidden layer (except the code): 64, 128
- Number of neurons the input layer as well as in the output layer: 13
- Number of neurons the code layer: 4
- Batch size: 32, 64
- Learning rate: 5%, 10%
- Patience for early stopping: 15
- Patience for learning rate reduction: 7
- Loss function: MSE
- Evaluation measure: MAE
- Optimizer: `Adam`
- Number of epochs: 75
- Batch normalization: between every layer

For training the SVMs (SVM and SVMclass), we perform a five-fold cross validation with hyperparameter tuning over the following grid:

- Kernel: `linear`, `radial`
- Gamma: 0.01, 0.1, 1
- Cost: 0.01, 1, 100

As regression mode of SVM we set the ε-regression with the `e1071` package default for ε. As classification mode for SVMclass we set the standard C-classification. We use the same grids for SVM and SVMclass.

The two models RF and RFclass are also trained using a five-fold cross validation with hyperparameter tuning. The hyperparameter grids are, just as above, the same for regression and classification:

- Number of variables considered for splits (.mtry): 1, 2, 3, 4, 5, 6
- Number of trees: 500, 1000

For the mtry parameter, there exists the rule of thumb to use $\lfloor \frac{p}{3} \rfloor$ for regression tasks and $\lfloor \sqrt{p} \rfloor$ for classification tasks with p being the number of explaining variables [52]. This is why we vary mtry around $\lfloor \frac{12}{3} \rfloor = 4$ resp. $\lfloor \sqrt{12} \rfloor = 3$.

F. The Algorithm

The bootstrapping, model training, and evaluation algorithm is depicted in the algorithm in Figure 3. All individual steps are described above. The algorithm is suitable for parallelization as the bootstrapping runs, i.e., the steps executed within the for-loop that begins in line 2 of Figure 3, are completely independent of each other.

In the hyperparameter optimization step (line 6) we use MSE as performance measure for SVM, the *Root Mean Squared Error* (RMSE) for RF, accuracy for RFclass and

```
begin
for i=1 to $n$ do
    begin
    Prepare datasets with $seed_i$ (split, manipulate, normalize);
    Train the two benchmark models on $\underline{train}$;
    Optimize hyperparameters of RNN, SVM, SVMclass, RF, RFclass, and NNA and retrain the best model in each case on $\underline{train}$;
    Measure all models' performance on $\underline{test}$;
    Measure detection performance of RNN, SVM, SVMclass, RF, RFclass, and NNA on $\underline{manip_{worst}}$ and on $\underline{manip_{random}}$;
    end
end
```

Figure 3. Procedure for model training and evaluation. Input: the original dataset; a seed vector $(seed_1, \ldots, seed_{100})$; four hyperparameter grids. Output: list of performance data.

the misclassification error (1-accuracy) for SVMclass. In the neural networks, the loss function is MSE, however, as performance metric for the hyperparameter optimization we use MAE. For RNN, the MAE is calculated on the target variable, for NNA, the MAE is calculated over all features. Note that these different performance measures are used for optimizing the hyperparameters but not for depicting the performance in Section VI-A.

As one can see from the algorithm, our approach is supervised when we train the models for predicting wine quality (LM, RNN, SVM, SVMclass, RF, and RFclass) resp. all features (BA, NNA). However, concerning a prediction of the manipulation label, our approach may be called semi-supervised. This is because we use labeled data to train the models, but only data that is labeled as "correct," i.e., that is not manipulated. Although in the analysis "correct" and "incorrect," i.e., manipulated data are used, no incorrect data are used for training—that is, one does not need a data set where "incorrect" data are already identified as incorrect. We use the information about which data entries are really "incorrect" only for the statistical analysis of the results for this paper.

G. Detection Performance

The detection performance is measured as follows: For RNN, SVM, SVMclass, RF, RFclass, we calculate the squared difference of the predicted quality and the given quality (which is possibly manipulated) for each data object (*Squared Error;* SE). Note that the results of the two classification models are treated as numeric values here. For NNA, we compute the detection performance in two different ways: on the one hand according to the regression models via the squared differences only on the target variable and on the other hand via the sum over the squared differences of all features (*Sum of Squared Errors;* SSE). In the following, when we distinguish between these two measurement methods, we denote with NNA the performance measure on only the target variable and with NNA_all the performance measure on all variables. Note that NNA and NNA_all are not two different kinds of models (unlike RF and RFclass resp. SVM and SVMclass) but denote only the two different kinds of detection performance measurement for the same model. For each model resp. each measurement type, we sort the data in descending order according to the respective deviation values. For example, for the first manipulation strategy, we map the manipulated test data to the following resorted sets:

$$\begin{aligned}\underline{manip_{worst}} \mapsto (&\underline{manip_{worst,RNN}},\\ &\underline{manip_{worst,NNA}},\\ &\underline{manip_{worst,NNA\_all}},\\ &\underline{manip_{worst,RF}},\\ &\underline{manip_{worst,RFclass}},\\ &\underline{manip_{worst,SVM}},\\ &\underline{manip_{worst,SVMclass}})\end{aligned}$$

.

Then, we determine the true/false positive rates when marking the first $q_i\%$ of the data objects in the sorted sets $\underline{manip_{worst,x}}$ with $x \in \{$RNN, NNA, NNA_all, RF, RFclass, SVM, SVMclass$\}$ as suspicious for $q_i = i$, $i = 1, 2, \ldots, 99$. The true positive rate tpr is defined as $tpr = TP/(TP + FN) = 1 - fnr$ and the false positive rate fpr is $fpr = FP/(TN + FP) = 1 - tnr$, where TP is the number of true positives, i.e., of manipulated objects that are marked suspicious, TN is the number of true negatives, i.e., of unmanipulated objects that are not marked, and FP and FN are the respective false positives/negatives and fnr and tnr the respective rates. If one would assign the "suspicious marks" randomly with equal probabilities to $q\%$ ($q \in [0, 100]$) of the data, the expected true/false positive rates would equal q, i.e., $\mathbb{E}[tpr] = \mathbb{E}[fpr] = q$, independent of the share of real positives/negatives. The values for $q = 0$ and $q = 100$ are meaningless since in the former case no object would be marked as suspicious and in the latter case all objects would be marked as suspicious.

To summarize the results of all runs, we calculate all quartiles of tpr and fpr for every q_i, i.e., minimum, first quartile, median, third quartile, and maximum. Before presenting the results of our analysis in Section VI, we describe how the set of possible hyperparameters via the sequential accumulative selection is found.

V. Hyperparameters for Neural Networks via Sequential Accumulative Selection

Since basically the set of possible hyperparameters especially for the neural network based models is infinite, it is quite natural that the size of this set has to be reduced. In doing so, for the neural networks, we perform the hyperparameter optimization in two steps. In the first step, we start with an initial set for possible hyperparameters, i.e., with a relatively large grid on the actually infinitely large space of hyperparameters. Additionally, we make an initial guess for a plausible setting as it is typical, for example, also for several optimization algorithms. In our case this means that each hyperparameter is initially set to a plausible value (underlined). This is done based on comparisons to similar problems as well as extensive trial-and-error pre-tests. We then reduce the size of the initial, large grid such that not all possible hyperparameter combinations need to be tried in the optimization step itself, which is performed as a grid search. We call the grid reduction "sequential accumulative selection."

For our case study, the initial hyperparameter grid for RNN is:

- Activation function: `linear`, `softplus`, $\underline{\texttt{ReLU}}$, `tanh`, `sigmoid`
- Number of hidden layers: 0, 1, 3, 5, $\underline{7}$
- Dropout rate: 0%, $\underline{5\%}$, 10%
- Number of neurons in each hidden layer: 32, 64, $\underline{128}$
- Batch size: $\underline{32}$, 64
- Learning rate: 5%, $\underline{10\%}$

The initial grid for NNA is:

- Activation function: `linear`, `softplus`, $\underline{\texttt{ReLU}}$, `tanh`, `sigmoid`
- Number of hidden layers (excluding the code layer): 0, 2, 4, $\underline{6}$
- Dropout rate: 0, $\underline{0.05}$, 0.1
- Number of neurons in each hidden layers (except the code layer): 32, 64, $\underline{128}$
- Batch size: $\underline{32}$, 64
- Learning rate: 0.05, $\underline{0.1}$

All other parameters are fixed to the values of Section IV-E. Note that we intentionally did not include varying numbers of neurons for the code layer of the autoencoder. This is because higher numbers of neurons in the code layer lead to a higher performance, but to a lower compression. Since both values are important for outlier detection, based on comparisons to similar examples, we chose four as a promising tradeoff.

Using the heuristic strategy of sequential accumulative selection, the two grids given above are thinned out so that the hyperparameter optimization in the algorithm in Figure 3 (in Section IV-F) performs within a reasonable runtime.

Next, we explain the sequential accumulative selection:

1) We start with performing 50 runs, i.e., on 50 different, randomly built training data sets, with the hyperparameters fixed to the underlined, plausible values except for the activation function, which is allowed to be any of the given possibilities. All activation functions that were taken at least once in the hyperparameter optimization in the 50 runs are declared to be also plausible, all others are deleted.
2) In the same fashion, the number of hidden layers is analyzed next, i.e., the hyperparameter optimizer has to optimize over the set of the plausible activation functions (due to step one there is possibly more than one plausible activation function) and the number of hidden layers. All values for the hidden layers that were chosen at least once are declared to be also plausible, all others are deleted. The plausible activation functions remain the same independent of whether they still appear in the set of optimal parameters of this second round.
3) This procedure is repeated in the following order with the number of neurons,
4) the dropout rate,
5) the batch size, and
6) the learning rate.

The results of the sequential accumulative selection, i.e., of the diminution of the possible hyperparameters, can be found in Section IV-E. For clear, the procedure of sequential accumulative selection is done separately for RNN and NNA and conducted on unmanipulated training data. The performance measure both for RNN and for NNA when selecting the best hyperparameter constellation in each run is the MAE, cf. IV-F.

Concerning the approach of the sequential accumulative selection, not only the choice of the plausible initial values for the specific hyperparameters is important, but also the order in which they are processed. Further, it is not clear in advance whether the grid is reduced at all. In the worst case, more models have to be trained than with simply using the initial, large grid. In our case, we had a distinct performance benefit through the sequential accumulative selection.

VI. Results

The results section is divided into three parts. First, we show the general model performances on the unmanipulated test data $\underline{test}$. That means, we analyze the ability of the regression models to predict correct quality values, the ability of the classification models to predict correct quality classes and the reconstruction ability of the autoencoders. Second, we examine the detection performance of the models except the benchmark models on the test data manipulated with manipulation strategy 1, i.e., on $\underline{manip_{worst}}$. Third, we investigate the detection performance under manipulation strategy 2, i.e., on $\underline{manip_{random}}$.

A. General Model Performance

The performance of all models, i.e., benchmark models LM and BA as well as the four resp. six more elaborate models SVM, SVMclass, RF, RFclass, RNN, NNA on the unmanipulated test data is depicted in Figure 4 through boxplots that capture the empirical distribution of the performance over all Monte Carlo runs. As measure for the predictive quality of all models we show the MAE where the predicted and the actual classes of the two classification models are treated as numeric values. For the two autoencoder models we show their performance on the target variable only (NNA and BA in Figure 4) and averaged over all variables (NNA_all and BA_all in Figure 4). The MAE values of the supervised models are normalized (the regression resp. classification is done on the unnormalized target feature) so that we can compare them to the autoencoder performance measurements.

As there are outliers in the performance of the neural network based models NNA and RNN, we truncated the

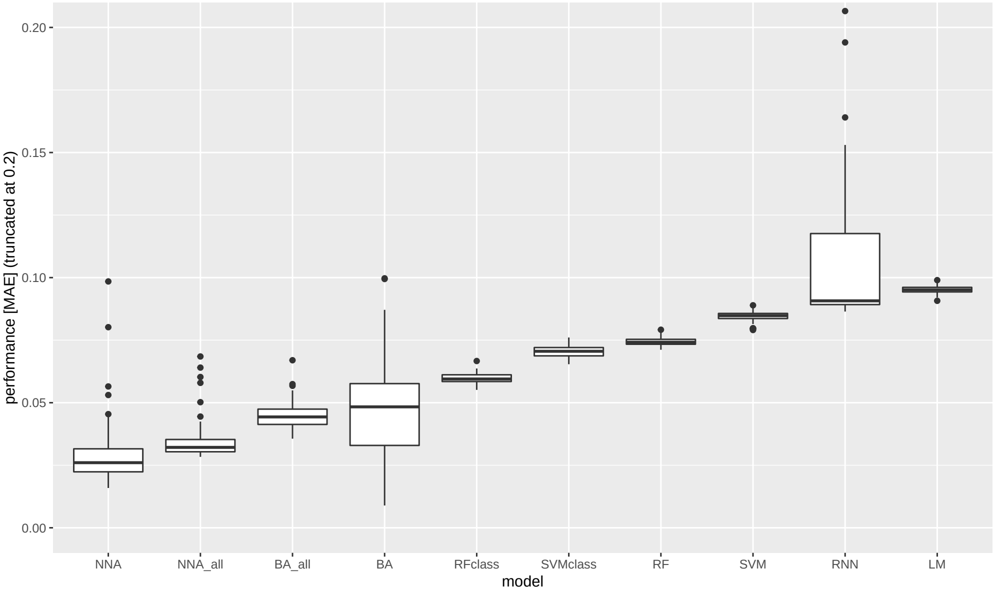


Figure 4. Boxplot of the performance (MAE) of all models (benchmark models and more elaborate models) on the unmanipulated test data. Outliers are truncated. In median, NNA is best, whilst the interquartile distance is the smallest for LM.

ordinate axis in Figure 4. The actual, rounded values of the five number summary, which is the basis for the boxplots drawn in Figure 4, is given in Table I.

TABLE I
FIVE NUMBER SUMMARY, I.E., MINIMUM, FIRST QUARTILE, MEDIAN, THIRD QUARTILE, AND MAXIMUM OF THE PREDICTIVE PERFORMANCE OF ALL MODELS.

	Min.	1st Qu.	Median	3rd Qu.	Max.
SVM	0.079	0.084	0.085	0.086	0.089
SVMclass	0.065	0.069	0.071	0.072	0.076
RF	0.071	0.073	0.074	0.075	0.079
RFclass	0.055	0.059	0.060	0.061	0.067
LM	0.091	0.094	0.095	0.096	0.099
BA	0.009	0.033	0.048	0.058	0.100
RNN	0.086	0.089	0.091	0.118	$6,106,068.000$
NNA	0.016	0.022	0.026	0.032	264.789
BA_all	0.036	0.041	0.044	0.047	0.067
NNA_all	0.028	0.030	0.032	0.035	242.428

Figure 4 and Table I show that NNA has the best predictive performance (in median) and that in general autoencoders, both NNA and BA, are better than the regression and classification models (in median). Note, however, that the application conditions for autoencoders and supervised methods are not the same. Autoencoders try to reconstruct the original input via compressing and decompressing. That particularly means that they know the value of the target variable (in terms of the supervised methods) as this is part of their input data also in the test set. It is therefore not clear whether the performance values in this section, although we calculate the MAE for all models, are really comparable between the autoencoders and the other, supervised methods.

We further see that the two classification models RFclass and SVMclass perform better than their corresponding regression models (not only in median, but also regarding the minimum and maximum values). RNN is (in median) better than the simplest model LM, however, its interquartile distance is the largest among all models, closely followed by that of BA evaluated only on the target variable. Concerning NNA, there is not much difference whether the model is evaluated only on the target variable or on all reconstructed variables. This is unlike BA, where we can see a larger variance when we evaluate only on the target variable.

Although there are slight performance differences, the SVM and RF models all show only a small distance between their minimum and maximum performance values, i.e., these models seem to be quite stable and well generalizing independent of the respective Monte Carlo run. It is easy to see that the neural network based models show higher performance variances than the other models, where NNA and especially RNN exhibit large to very large outliers, i.e., really bad performing models. Regarding the exorbitantly high maximum value of RNN, we had a closer look on the respective Monte Carlo run. This high MAE is driven by one (an extremely bad one) of the 1950 predictions in the respective test set. Looking deeper at the corresponding wine features we see that this extremely bad prediction belongs to a wine with by far the highest value of “free sulfur dioxide” in the test set (two times as high as the wine with the second highest value) and the highest value of “total sulfur dioxide”. With an unlucky weighting of these features in the neural network, the extreme prediction value may be explained.

Regarding the times for hyperparameter optimization (not considering the sequential accumulative selection for RNN and NNA and not considering BA and LM here, as we did not optimize any hyperparameters for these two models) and training, we see the average of each model depicted in Table II. By far, the optimization and training time of RNN is the longest, where the fastest is that of RFclass. However, the grid for RNN is also larger than that of RF(class), thus, a mere comparison of the training times is not meaningful just like that. But keeping in mind that despite the relatively small grid, RF(class) performs better than RNN, RNN is no good choice for predicting the wine quality. It may be the case that other neural network architectures not captured by our hyperparameter grid would show a better performance than our RNN.

TABLE II
AVERAGE DURATION OF HYPERPARAMETER OPTIMIZATION AND TRAINING OF THE RESPECTIVE MODELS MEASURED IN MINUTES.

Model	Training Time (in minutes)
SVM	16.49
SVMclass	63.89
RF	16.79
RFclass	5.73
RNN	134.06
NNA	15.41

B. Manipulation Detection Performance

Next, we evaluate the detection performance of the non benchmark models. For this, we do set neither an explicit threshold for the share of data objects to be marked as suspicious nor an explicit threshold for the SE resp. SSE beyond which the data objects have to be marked as suspicious since the aim of this work is not to find a classifier for manipulated wine data quality but the comparison of the four resp. six models: SVM, SVMclass, RF, RFclass, RNN, NNA, where NNA's detection performance may be measured in two different ways (as before considering only the target feature or all reconstructed features). How a threshold can be found is, e.g., outlined in [53]. We first show and discuss the results of the first manipulation strategy with the corresponding test set $\underline{manip_{worst}}$ and then continue with the second manipulation strategy with the corresponding test set $\underline{manip_{random}}$. Note that in contrast to the general model performance analyzed in Section VI-A, the detection performance is unquestionably comparable between the autoencoder and the other models as we work with orderings here and not with absolute values, as it is the case when, e.g., comparing several MAE values. All statements in this section apply in tendency, as they depend on chance (especially the neural networks) and on q.

1) First Manipulation Strategy $manip_{worst}$: To illustrate the detection performance of the models mentioned above in the context of the manipulation of the 10% worst rated wines, we calculate tpr and fpr for all Monte Carlo like runs and for all $q_i = 1, \dots, 99$. For all q_i, we calculate the five quartiles of tpr and fpr for each model and plot these values against q. The results are depicted in Figures 5 (tpr) and 6 (fpr).

As we can easily observe in Figure 5, all models are better than randomly guessing since all lines are above the diagonal. An optimal detection model would linearly increase and reach a tpr of 100% at $q = 10\%$, as already explained in Section IV-G. Among all models, the NNA (and NNA_all) is by far the worst model concerning the detection of the manipulated wines. This is interesting as NNA is, at the same time, the model performing best on the unmanipulated test data. Further, autoencoders are generally used for outlier detection, but here it seems that the regression models are more suitable for the detection of data manipulation in the target variable. The fpr in Figure 6 shows the corresponding behavior of the models. Here, an optimal model would have a fpr of 0 up to $q = 10\%$, i.e., there are no false positives among the first 10% of the data objects when sorted according to their deviance of predicted and actual (manipulated) values, and then linearly increase to the point $(100\%, 100\%)$. A summary of the medians of all models and all manipulation strategies at $q = 10\%$ is given in Table III.

2) Second Manipulation Strategy $manip_{random}$: We repeat the same analysis as in Section VI-B1 for the second manipulation strategy, where we randomly changed the target variable of 10% of the data to a plausible, but also random value. In Figures 7 and 8 we show the tpr resp. fpr on the manipulated test set $\underline{manip_{random}}$. Note that among the 10% data objects marked as manipulated, not all of them are necessarily manipulated. This is why we slightly adjust the analysis resp. the manipulation markings of the detection performance by marking only those data objects as manipulated where a manipulation has actually taken place ($\underline{manip_{random}} \mapsto \underline{manip_{random2}}$). This leads to a maximum manipulation rate of 10%, i.e., the actual manipulation rate is somewhere between 0% and 10% and depends on the respective Monte Carlo run.

The adjustment of the analysis is depicted in Figures 9 and 10 and works, as can be seen in the figures, in favor of the models, i.e., the adjustment increases their tpr and lowers their fpr. In the further course, we only discuss the results of the adjusted detection performance measurements.

When regarding Figure 9, it is immediately obvious that all models have more trouble detecting the manipulated data objects than it was the case for the first manipulation strategy. The order of the models concerning their detection ability is more or less the same as in Section VI-B1 but at a lower level. The best model shows a median tpr of about 37% for $q = 10\%$ where on $\underline{manip_{worst}}$ the median tpr at $q = 10\%$ is about 87% (see also Table III). Again, the NNA is the worst performing model, where its minimum tpr over all runs when measuring its detection performance only on the target variable is even worse than randomly guessing. Note that on $\underline{manip_{random2}}$, an optimal model would reach a tpr of 100% at the latest for $q = 10\%$ and possibly already for $q < 10\%$.

VII. CONCLUSION

We analyze the ability of different machine learning methods for detecting manipulated wine ratings. In detail, we

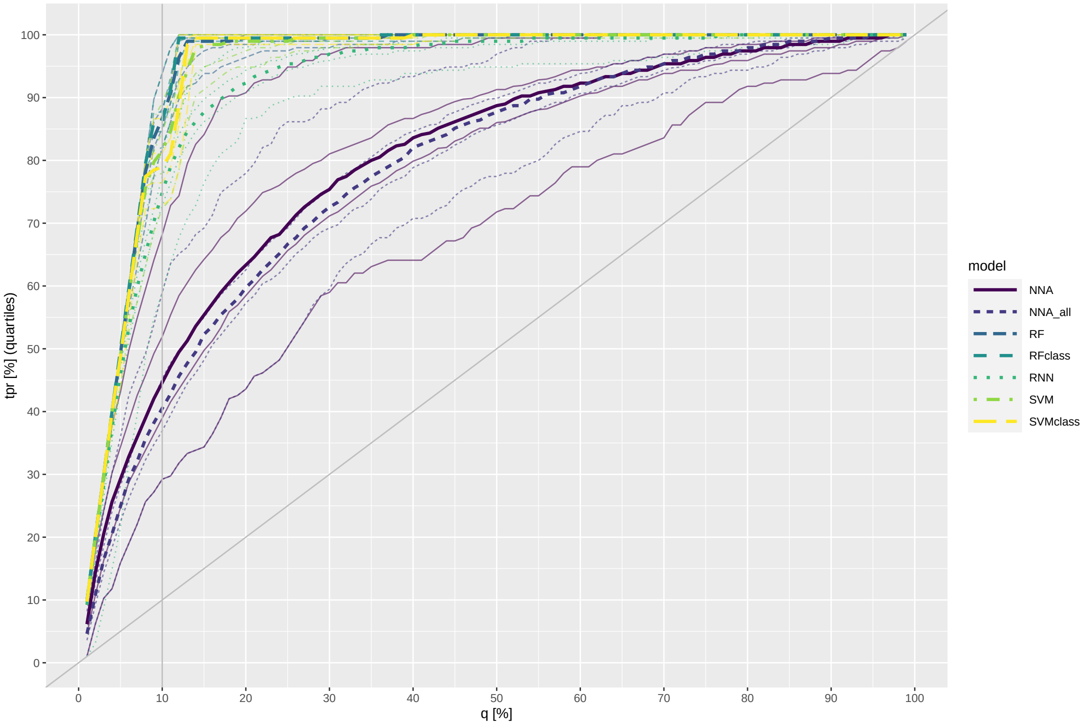


Figure 5. The five quartiles of tpr on $\underline{manip_{worst}}$ for varying q for all models distinguishable by color and line type. The median is drawn thicker. Additionally, the diagonal and the 10% line are depicted.

TABLE III
MEDIAN DETECTION PERFORMANCE IN % OF ALL MODELS ON $\underline{manip_{worst}}$ (w) AND $\underline{manip_{random}}$ (r) RESP. $\underline{manip_{random2}}$ $(r2)$ FOR $q = 10\%$.

	tpr,w	tpr,r	tpr,$r2$	fpr,w	fpr,r	fpr,$r2$
RF	86.667	26.154	36.965	1.425	8.148	8.013
RFclass	86.154	24.615	33.835	1.481	8.319	8.256
SVM	81.538	23.077	32.046	1.994	8.490	8.329
SVMclass	78.974	21.538	29.720	2.279	8.661	8.551
RNN	74.872	21.538	29.458	2.735	8.661	8.505
NNA	44.615	15.385	18.919	6.097	9.345	9.304
NNA_all	40.513	14.359	16.288	6.553	9.459	9.491

consider autoencoders implemented via neural networks, neural network regressions, support vector machines used both for regression and classification, as well as random forests used both for regression and classification. We measure these models' general performance on an unmanipulated test set and their detection performance on two different manipulated test sets. The first manipulation strategy increases the quality ratings of those wines that were originally the worst rated. The second manipulation strategy changes the quality ratings of randomly picked wines to a value drawn from the empirical distribution of the remaining, unmodified part of the test data. As a benchmark for the models' general performance we additionally consider a linear regression and a minimalistic benchmark autoencoder. The latter two models are not used for the manipulation detection. All data splitting, training, and testing steps are repeated 100 times in a Monte Carlo like manner in order to get more robust results. Our case study is conducted on two vinho verde datasets.

We find that the more elaborate models generally perform better than their respective benchmark model. The autoencoders show the smallest mean absolute error (in median over all runs) on the unmanipulated test data, but it is not clear whether the performance measures are really comparable since the prerequisites for autoencoders and supervised models are not the same. Among the supervised models, the support vector model classification and the random forest classification (their results also measured via the mean absolute error) basically perform best. The neural network based models show a great variability across the different runs.

Concerning the models' ability to detect manipulated wine ratings, the supervised machine learning models outperform the autoencoder. For manipulation strategy one, the random forest models show a true positive rate of over 86% (in median) when marking the 10% test wines where the predicted quality and the (possibly manipulated) actual quality deviate most. As expected, all models show a better detection performance on manipulation strategy one (which we intended to be economically reasonable) than on manipulation strategy two (the random manipulation). Since the classification random forest is among the models with the best performance values and has the least optimization and training time (where the hyperparameter grid we use is quite sparse), it may be denoted as the most suitable model for the detection of manipulated

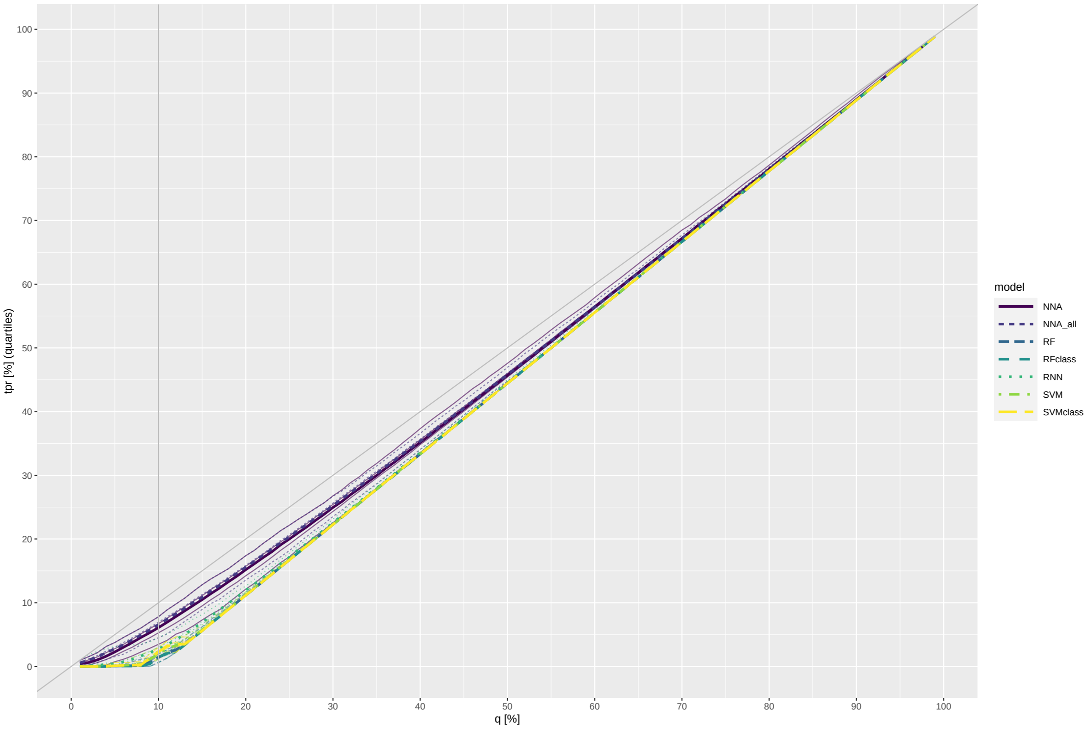


Figure 6. The five quartiles of fpr on $\underline{manip_{worst}}$ for varying q for all models distinguishable by color and line type. The median is drawn thicker. Additionally, the diagonal and the 10% line are depicted.

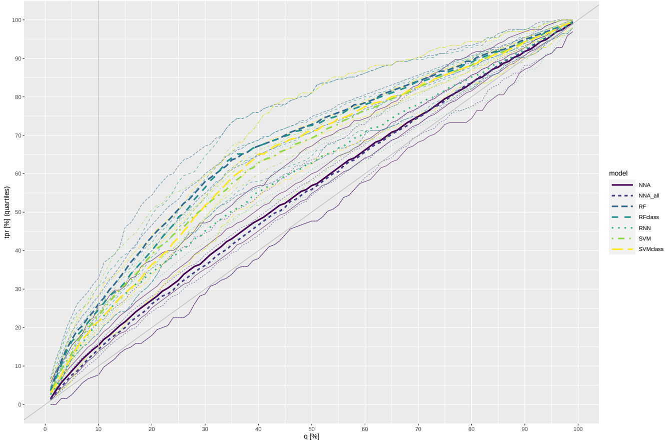


Figure 7. The five quartiles of tpr on $\underline{manip_{random}}$ for varying q for all models distinguishable by color and line type. The median is drawn thicker. Additionally, the diagonal and the 10% line are depicted.

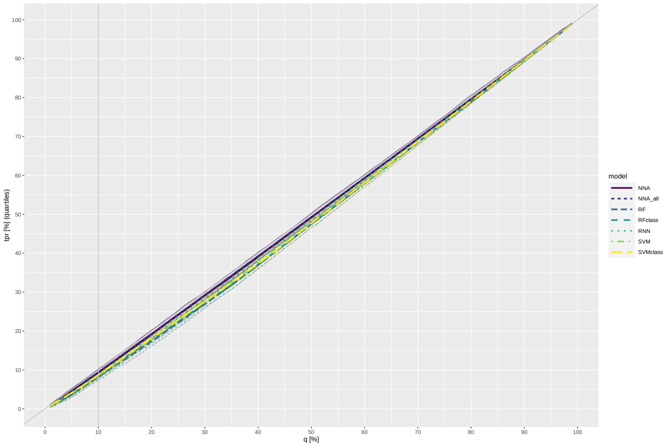


Figure 8. The five quartiles of fpr on $\underline{manip_{random}}$ for varying q for all models distinguishable by color and line type. The median is drawn thicker. Additionally, the diagonal and the 10% line are depicted.

wine ratings.

In order to allow for a relatively large initial hyperparameter grid for the neural network based models (autoencoder network and regression neural network), we establish the procedure of sequential accumulative selection, where in a pre step the initial grid is sequentially reduced before the actual full grid search is applied. Despite the comparably large effort we put into the optimization of the neural networks, they show a great variability.

VIII. Future Work

In this paper, we first assume that it is reasonable that manipulations are applied to low rated wines to make them appear better to increase sales numbers. As a reference setting, we additionally apply a random manipulation. However, it would be interesting to test our approach also on other, somehow meaningful manipulation strategies, including, e.g., intentional and unjustified down ratings. Future work could also deal with the detection of faked ratings when there are multiple ratings

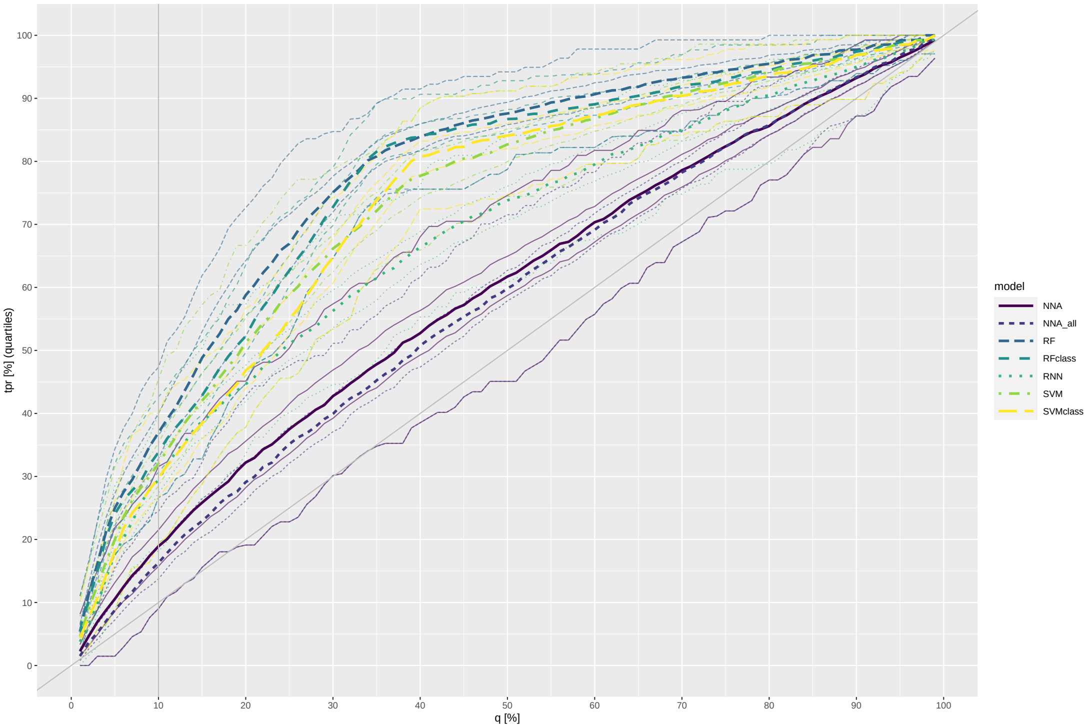


Figure 9. The five quartiles of tpr on $\underline{manip_{random2}}$ considering the actually manipulated data objects for varying q for all models distinguishable by color and line type. The median is drawn thicker. Additionally, the diagonal and the 10% line are depicted.

per product as it is typical for many online stores or rating portals. Are there ways to detect the faked/manipulated ratings (whether better or worse) when there are many ratings for the same product? In this context, many stores and portals offer the possibility to write a review in addition to the plain rating. The processing of such information (via natural language processing) is likely to be useful here. In our approach, as it is typical for wine and other online ratings, we did not assume that cryptographic means for the prevention of manipulations exist. This is mainly because ratings can be manipulated before they are written down the first time. However, it could be interesting to check how the introduction of some cryptographic means, e.g., manipulation detection codes [55], which are some kind of checksums, could influence the detection of manipulated ratings.

Of course, other application areas apart from wine can be investigated with our approach, for example, ratings for products in online stores, restaurants, hotels. The detection of fraud in telecommunication, insurance, etc. [54] is also closely related. It could be of interest to identify the similarities and differences between these applications and how they should be addressed. When analyzing wine ratings, it would also be interesting to transfer our approach to other, larger datasets with more features, such as countries, producing regions, price segments, etc. and analyze the stability of the models' performances.

The procedure of *sequential accumulative selection* (as explained in Section V) can be further analyzed. One might investigate whether and how the order of the features is important. Comparisons to other hyperparameter selection methods are also possible (cf. [32]).

Last but not least, it should be noted that the topic of explainable AI and responsible AI is rapidly growing in importance [56]. It would be helpful to get to know the reasons why a manipulation detection model marks a certain data object as suspicious. As few as possible false positives are to be marked, whereas all manipulated ones are to be recognized if possible. So how can the decisions of the recognition algorithms be (understandably) explained?

Acknowledgment

Michaela Baumann is with NÜRNBERGER Versicherung, Germany. The opinions expressed here are her own and not necessarily those of her employer.

The authors thank Lars Grüne and Bernhard Herz, both with University of Bayreuth, Germany.

References

[1] M. Baumann and M. H. Baumann, "Autoencoder vs. Regression Neural Networks for Detecting Manipulated Wine Ratings," The Seventeenth International Multi-Conference on Computing in the Global Information Technology (ICCGI), 2022, pp. 7-13

[2] G. Rosso, Italian Wines 2021 *(English Edition)*, Gambero Rosso, 2021

[3] R. Parker, The Wine Advocate, https://www.robertparker.com/articles/the-wine-advocate, accessed: 2022.12.08

[4] Gault&Millau, https://www.gaultmillau.com/, accessed: 2022.12.08

[5] Guide Michelin, https://guide.michelin.com/en, accessed: 2022.12.08

[6] D. Freedman, R. Pisani, and R. Purves, "Statistics," 4th ed., W. W. Norton & Company, Inc., New York, London, 2007, Chapters 10-12

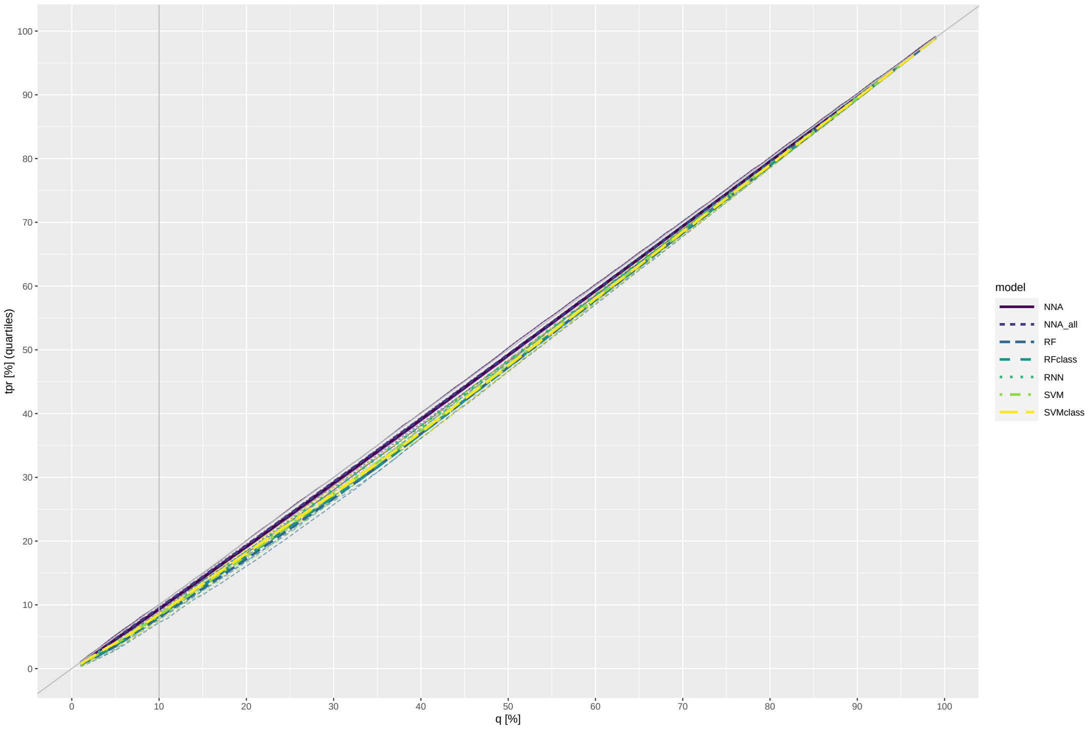


Figure 10. The five quartiles of fpr on $\underline{manip_{random2}}$ considering the actually manipulated data objects for varying q for all models distinguishable by color and line type. The median is drawn thicker. Additionally, the diagonal and the 10% line are depicted.

[7] E. Gelenbe, Z. H. Mao, and Y. D. Li, “Function Approximation with Spiked Random Networks,” in IEEE Transactions on Neural Networks, vol. 10, no. 1, 1999, pp. 3-9

[8] E. Gelenbe, “Random Neural Networks with Negative and Positive Signals and Product Form Solution,” in Neural Computataion, vol. 1, no. 4, 1989, pp. 502-510

[9] T. Poggio, H. Mhaskar, L. Rosasco, B. Miranda, and Q. Liao, “Why and When Can Deep-but Not Shallow-networks Avoid the Curse of Dimensionality: A Review,” in International Journal of Automation and Computing, vol. 14, no. 5, 2017, pp. 503-519

[10] Y. LeCun, Y. Bengio, and G. Hinton, “Deep Learning,” in Nature, vol. 521, no. 7553, 2015, pp. 436-444

[11] H. Drucker, C. J. Burges, L. Kaufman, A. Smola, and V. Vapnik, “Support Vector Regression Machines,” in Advances in Neural Information Processing Systems, vol. 9, 1996

[12] I. Steinwart and A. Christmann, “Support Vector Machines,” Springer, 2008

[13] L. Breiman, “Bagging Predictors,” in Machine Learning, vol. 24, 1996, pp. 123-140

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

[15] S. Hawkins, H. He, G. Williams, and R. Baxter, “Outlier Detection Using Replicator Neural Networks,” Data Warehousing and Knowledge Discovery, 2002, pp. 170-180

[16] M. Sakurada and T. Yairi, “Anomaly Detection Using Autoencoders with Nonlinear Dimensionality Reduction,” Proceedings of the MLSDA 2014 2nd Workshop on Machine Learning for Sensory Data Analysis, 2014, pp. 4-11

[17] G. E. Hinton and R. R. Salakhutdinov, “Reducing the Dimensionality of Data with Neural Networks,” in Science, vol. 313, no. 5786, 2006, pp. 504-507

[18] J. D. Kelleher, “Deep Learning,” MIT press, 2019

[19] M. A. Kramer, “Autoassociative Neural Networks,” in Computers & Chemical Engineering, vol. 16, no. 4, 1992, pp. 313-328

[20] M. A. Kramer, “Nonlinear Principal Component Analysis Using Autoassociative Neural Networks,” in AIChE journal, vol. 37, no. 2, 1991, pp. 233-243

[21] D. L. Donoho, “High-Dimensional Data Analysis: The Curses and Blessings of Dimensionality,” in AMS math challenges lecture, 2000

[22] J. W. Tukey, “The Future of Data Analysis,” in The Annals of Mathematical Statistics, vol. 33, no. 1, 1962, pp. 1-67

[23] P. Cortez, A. Cerdeira, F. Almeida, T. Matos, and J. Reis, “Modeling Wine Preferences by Data Mining from Physicochemical Properties,” in Decision Support Systems, vol. 47, no. 4, 2009, pp. 547-553

[24] P. Cortez et al., “Using Data Mining for Wine Quality Assessment,” in International Conference on Discovery Science, Springer, Berlin, Heidelberg, 2009, pp. 66-79

[25] Y. Gupta, “Selection of Important Features and Predicting Wine Quality Using Machine Learning Techniques,” in Procedia Computer Science, vol. 125, 2018, pp. 305-312

[26] À. Nebot, F. Mugica, and A. Escobet, “Modeling Wine Preferences from Physicochemical Properties Using Fuzzy Techniques,” in SIMULTECH, 2015, pp. 501-507

[27] S. Kumar, Y. Kraeva, R. Kraleva, and M. Zymbler, “A Deep Neural Network Approach to Predict the Wine Taste Preferences,” in Intelligent Computing in Engineering, Springer, Singapore, 2020, pp. 1165-1173

[28] P. Abbal, J. M. Sablayrolles, E. Matzner-Lober, and A. Carbonneau, “A Model for Predicting Wine Quality in a Rhône Valley Vineyard,” in Agronomy Journal, vol. 111, no. 2, 2019, pp. 545-554

[29] O. Ashenfelter, “Predicting the Quality and Prices of Bordeaux Wine,” in The Economic Journal, vol. 118, no. 529, 2008, pp. F174-F184

[30] O. Ashenfelter, “Predicting the Quality and Prices of Bordeaux Wine,” in Journal of Wine Economics, vol. 5, no. 1, 2010, pp. 40-52

[31] R. Schwarz, “Predicting Wine Quality from Terrain Characteristics with Regression Trees,” in Cybergeo: European Journal of Geography, 1997

[32] T. H. Y. Chiu, C. Wu, and C. H. Chen, “A Generalized Wine Quality Prediction Framework by Evolutionary Algorithms,” in International Journal of Interactive Multimedia & Artificial Intelligence, vol. 6, no. 7, 2021, pp. 60-70

[33] R. Andonie, A. M. Johansen, A. L. Mumma, H. C. Pinkart, and S. Vajda, “Cost Efficient Prediction of Cabernet Sauvignon Wine Quality,” IEEE Symposium Series on Computational Intelligence (SSCI), 2016, pp. 1-8

[34] U. G. Mahima, Y. Patidar, A. Agarwal, and K. P. Singh, “Wine Quality Analysis Using Machine Learning Algorithms,” Micro-Electronics and Telecommunication Engineering, Springer, 2020, pp. 11-18

[35] S. Bhattacharjee and M. R. Chaudhuri, “Understanding Quality of Wine Products Using Support Vector Machine in Data Mining,” in Prestige International Journal of Management & IT-Sanchayan, vol. 5, no. 1, 2016, pp. 67-80

[36] S. Kumar, K. Agrawal, and N. Mandan, “Red Wine Quality Prediction Using Machine Learning Techniques,” International Conference on Computer Communication and Informatics (ICCCI), 2020, pp. 1-6

[37] V. Chandola, A. Banerjee, and V. Kumar, “Anomaly Detection: A Survey”, ACM Comput. Surv., vol. 41, no. 3, 2009, article no. 15, pp. 1-15

[38] V. Hodge and J. Austin, “A Survey of Outlier Detection Methodologies.” Artificial Intelligence Review, vol. 22, 2004, pp. 85-126

[39] M. H. Bhuyan, D. K. Bhattacharyya, and J. K. Kalita, “Network Anomaly Detection: Methods, Systems and Tools,” in IEEE Communications Surveys & Tutorials, vol. 16, no. 1, 2014, pp. 303-336

[40] A. Patcha and J.-M. Park, “An Overview of Anomaly Detection Techniques: Existing Solutions and Latest Technological Trends,” in Computer Networks, vol. 51, no. 12, 2007, pp. 3448-3470

[41] R. Chalapathy and S. Chawla, “Deep Learning for Anomaly Detection: A Survey,” preprint on arXiv, https://arxiv.org/abs/1901.03407, 2019

[42] C. C. Noble and D. J. Cook, “Graph-Based Anomaly Detection,” Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2003, pp. 631-636

[43] Wine Quality Datasets, Universidade do Minho, http://www3.dsi.uminho.pt/pcortez/wine/, accessed: 2022.12.08

[44] kaggle (Piyush Agnihotri), White Wine Quality—Simple and Clean Practice Dataset for Regression or Classification Modelling, https://www.kaggle.com/piyushagni5/white-wine-quality, accessed: 2022.12.08

[45] kaggle (UCI Machine Learning), Red Wine Quality—Simple and Clean Practice Dataset for Regression or Classification Modelling, https://www.kaggle.com/uciml/red-wine-quality-cortez-et-al-2009, accessed: 2022.12.08

[46] Open Data Commons—Legal Tools for Open Data, Database Contents License (DbCL) v1.0, https://opendatacommons.org/licenses/dbcl/1-0/, accessed: 2022.12.08

[47] T. Shin, “Predicting Wine Quality with Several Classification Techniques” Towards Data Science, 2020, https://towardsdatascience.com/predicting-wine-quality-with-several-classification-techniques-179038ea6434, accessed: 2022.12.08

[48] D. Nguyen, “Red Wine Quality Prediction Using Regression Modeling and Machine Learning,” Towards Data Science, 2020, https://towardsdatascience.com/red-wine-quality-prediction-using-regression-modeling-and-machine-learning-7a3e2c3e1f46, accessed: 2022.12.08

[49] F. Rodríguez Mir, “Red Wine Quality,” Data UAB, 2019, https://datauab.github.io/red_wine_quality/, accessed: 2022.12.08

[50] *unknown,* “Wine Quality Prediction,” cppsecrets.com, 2021, https://cppsecrets.com/users/10126100104105114971061121141111061019964103109971051084699111109/WINE-QUALITY-PREDICTION.php, accessed: 2022.12.08

[51] D. Alekseeva, “Red and White Wine Quality,” RPubs, https://rpubs.com/Daria/57835, accessed: 2022.12.08

[52] M. Hatz, “Der Einfluss von mtry auf Random Forests” (in English: “The Influence of mtry on Random Forests”), Master’s Thesis, 2018

[53] N. Japkowicz, C. Myers, and M. Gluck, “A Novelty Detection Approach to Classification,” in IJCAI, vol. 1, 1995, pp. 518-523

[54] M. Baumann, “Improving a Rule-based Fraud Detection System with Classification Based on Association Rule Mining,” INFORMATIK, 2021, pp. 1121-1134

[55] R. R. Jueneman, “A High Speed Manipulation Detection Code,” in Advances in Cryptology — CRYPTO’ 86, Lecture Notes in Computer Science, vol. 263, 1987, pp. 327-346

[56] M. Baumann, “Data Science Challenge 2021: Explainable Machine Learning,” https://github.com/DeutscheAktuarvereinigung/Data-Science-Challenge2021_Explainable-Machine-Learning, accessed: 2022.12.08

An Investigation of Twitter Users Who Gave Likes to Deleted Tweets Disclosing Submitters' Personal Information

Yasuhiko Watanabe, Toshiki Nakano, Hiromu Nishimura, and Yoshihiro Okada
Ryukoku University
Seta, Otsu, Shiga, Japan
Email: watanabe@rins.ryukoku.ac.jp, t180450@mail.ryukoku.ac.jp,
t160405@mail.ryukoku.ac.jp, okada@rins.ryukoku.ac.jp

Abstract—**Nowadays, many people use a Social Networking Service (SNS). Most SNS users are careful in protecting the privacy of personal information: name, age, gender, address, telephone number, birthday, etc. However, some SNS users disclose their personal information that can threaten their privacy and security even if they use unreal name accounts. In this study, we investigated Twitter users who gave likes to tweets disclosing submitters' personal information that potentially threatened submitters' privacy and security. We collected 318 tweets promising to disclose submitters' personal information. 30 tweets out of them were deleted in short order, specifically within a week after they were submitted. Then, we investigated the relations between the submitters of these 318 tweets and users who gave likes to them, taking into account whether or not the submitters deleted their tweets in short order. The results of our survey showed that the submitters followed most of the users mutually before the users gave likes to the tweets promising to disclose submitters' personal information whether or not the submitters deleted their tweets in short order. On the other hand, most of the users did not follow each other although they followed the same submitters and gave likes to their tweets.**

Keywords–personal information; Twitter; SNS; mutual follows; privacy risk; unreal name account user.

I. Introduction

Nowadays, many people use a Social Networking Service (SNS) to communicate with each other and try to enlarge their circle of friends. SNS users are generally concerned about potential privacy risks. To be specific, they are afraid that unwanted audiences will obtain information about them or their families, such as where they live, work, and play. As a result, SNS users are generally careful in disclosing their personal information. However, some SNS users, especially young users, disclose their personal information on their profiles, for example, real full name, gender, hometown and full date of birth, which can potentially be used to identify details of their real life. In order to discuss the reasons why some SNS users disclose their personal information willingly, it is important to investigate who they want to read their SNS messages disclosing their personal information. However, it is difficult to ask them who they want to read them. To solve this problem, it is important to investigate who gave responses to their SNS messages disclosing their personal information. This is because, if submitters felt unwanted audiences read and gave responses to their SNS messages disclosing their personal information, they would delete them. In order to investigate who gave responses to SNS messages disclosing submitters' personal information, we investigated Twitter users who gave likes to tweets disclosing submitters' personal information [1]. Furthermore, we investigated whether users concerned with a tweet disclosing submitter's personal information followed each other. In other words, we investigated

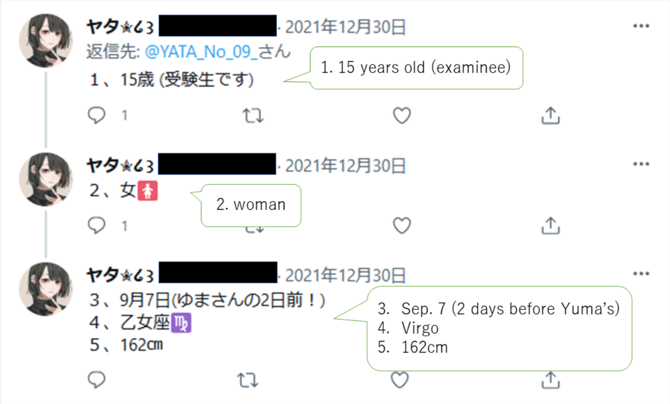
- whether a submitter followed users who gave likes to his/her tweet disclosing his/her personal information,
- whether users who gave likes to a tweet disclosing submitter's personal information followed the submitter, and
- whether each user who gave a like to a tweet disclosing submitter's personal information followed every other user who gave a like to the same tweet.

In this study, we examine these points by checking their Twitter follow relations, taking into account whether or not a submitter deleted his/her tweet disclosing his/her personal information in short order. The investigation is based on an idea: when an user follow someone on Twitter, he/she is not a stranger to the user. By using the results of the investigation, we discuss the relations of submitters of tweets disclosing their personal information and users who gave likes to the tweets. The results of the investigation might improve social media design elements, such as privacy controls and friend introductions.

The rest of this paper is organized as follows: in Section II, we survey the related works. In Section III, we show how to collect tweets where submitters seemingly disclosed their personal information honestly and detect users who gave likes to them. In Section IV, we examine whether users concerned with a tweet disclosing submitter's personal information followed each other and discuss the relations of them. In Section V, we investigate users who gave likes to tweets that disclosed submitters' personal information and were deleted in short order. Finally, in Section VI, we present our conclusions.

II. Related work

Personally identifiable information is defined as information, which can be used to distinguish or trace an individual's identity, such as social security number, biometric records, etc. alone, or when combined with other information that is linkable to a specific individual, such as date and place of birth, mother's maiden name, etc. [2] [3]. Internet users are generally concerned about unwanted audiences obtaining personal information. Fox et al. reported that 86% of Internet users are concerned that unwanted audiences will obtain information about them or their families [4]. However, Internet users, especially young users, tend to disclose personal information on their profiles, for example, real full name, gender,



ヤタ⭐︎63 [redacted] · 2021年12月30日

返信先: @YATA_No_09_さん

1、15歳 (受験生です)

1. 15 years old (examinee)

ヤタ⭐︎63 [redacted] · 2021年12月30日

2、女🚺

2. woman

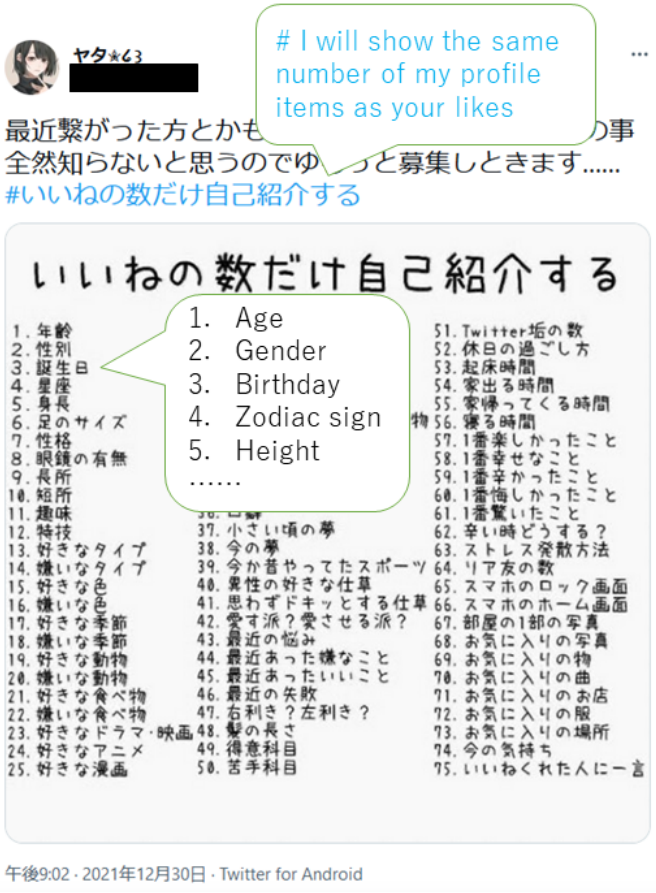
ヤタ⭐︎63 [redacted] · 2021年12月30日

3、9月7日(ゆまさんの2日前！)
4、乙女座♍
5、162㎝

3. Sep. 7 (2 days before Yuma's)
4. Virgo
5. 162cm

Figure 1. An unreal name account user, *Yata*, disclosed her personal profile items in her tweets.

hometown and full date of birth. As a result, many researchers discussed the reasons why young users willingly disclose personal information on their SNS profiles. Acquisti and Gross explained this phenomenon as a disconnection between the users' desire to protect their privacy and their actual behavior [5]. Also, Livingstone pointed out that teenagers' conception of privacy does not match the privacy settings of most SNSs [6]. On the other hand, Barnes argued that Internet users, especially teenagers, are not aware of the nature of the Internet and SNSs [7]. Viseu, Clement, and Aspinall reported that many online users believe the benefits of disclosing personal information in order to use an Internet site is greater than the potential privacy risks [8]. The authors think that most SNS users are seriously concerned about their privacy and security. However, they often underestimate the risk of their online messages and submit them. Hirai reported that many users had troubles in SNSs because they never thought that strangers observed their communication with their friends [9]. Watanabe, Nishimura, Chikuki, Nakajima, and Okada reported that some Twitter users submitted tweets disclosing their personal information that can threaten their privacy and security even if they use unreal name accounts [10]. In this study, we investigate what relations existed between users concerned with a tweet disclosing submitter's personal information. In order to analyze relations in communities, many researchers have adopted tie strength. Granovetter defined tie strength as the strength of a friendship: close friends are strong ties and acquaintances are weak ties [11]. Both strong ties and weak ties are useful because they provide access to different types of resources [12]. For example, strongly tied peers have greater motivation for assistance and provide access to information known by the group [11]. In contrast, weak ties provide diverse perspectives as well as novel information and resources [13]. Panovich, Miller, and Karger investigated the relation of tie strength to answer quality and showed that social network Q&A is more effective when the asker and answerer know each other well [14]. Gilbert and Karahalios proposed a predictive model of tie strength on Facebook using profile characteristics [15]. In this study, we investigate what relations existed between Twitter users concerned with a tweet disclosing submitter's personal information by checking their Twitter follow relations.



ヤタ⭐︎63
[redacted]

I will show the same number of my profile items as your likes

最近繋がった方とかも の事
全然知らないと思うのでゆ っと募集しときます……
#いいねの数だけ自己紹介する

いいねの数だけ自己紹介する

1. Age
2. Gender
3. Birthday
4. Zodiac sign
5. Height
……

1. 年齢
2. 性別
3. 誕生日
4. 星座
5. 身長
6. 足のサイズ
7. 性格
8. 眼鏡の有無
9. 長所
10. 短所
11. 趣味
12. 特技
13. 好きなタイプ
14. 嫌いなタイプ
15. 好きな色
16. 嫌いな色
17. 好きな季節
18. 嫌いな季節
19. 好きな動物
20. 嫌いな動物
21. 好きな食べ物
22. 嫌いな食べ物
23. 好きなドラマ･映画
24. 好きなアニメ
25. 好きな漫画

37. 小さい頃の夢
38. 今の夢
39. 今か昔やってたスポーツ
40. 異性の好きな仕草
41. 思わずドキッとする仕草
42. 愛す派？愛させる派？
43. 最近の悩み
44. 最近あった嫌なこと
45. 最近あったいいこと
46. 最近の失敗
47. 右利き？左利き？
48. 髪の長さ
49. 得意科目
50. 苦手科目

51. Twitter垢の数
52. 休日の過ごし方
53. 起床時間
54. 家出る時間
55. 家帰ってくる時間
56. 寝る時間
57. 1番楽しかったこと
58. 1番幸せなこと
59. 1番辛かったこと
60. 1番悔しかったこと
61. 1番驚いたこと
62. 辛い時どうする？
63. ストレス発散方法
64. リア友の数
65. スマホのロック画面
66. スマホのホーム画面
67. 部屋の1部の写真
68. お気に入りの写真
69. お気に入りの物
70. お気に入りの曲
71. お気に入りのお店
72. お気に入りの服
73. お気に入りの場所
74. 今の気持ち
75. いいねくれた人に一言

午後9:02 · 2021年12月30日 · Twitter for Android

Figure 2. A tweet promising to disclose the same number of submitter's personal profile items as likes to it.

III. A COLLECTION OF TWEETS DISCLOSING SUBMITTERS' PERSONAL INFORMATION

It is difficult to collect tweets disclosing submitters' personal information, such as tweets in Figure 1, directly. To solve this problem, we focused on tweets where submitters promised their audiences to disclose the same number of their own personal profile items as likes to their tweets. Figure 2 shows a tweet submitted by *Yata* on December 30, 2021. Both in Figure 1 and Figure 2, her screen name is redacted for privacy. Figure 2 shows that *Yata* promised her audiences to disclose the same number of her personal profile items as likes to her tweet. Actually, as shown in Figure 1, *Yata* submitted three replies disclosing her five personal profile items to her tweet shown in Figure 2 on December 30, 2021. Watanabe, Nishimura, Chikuki, Nakajima, and Okada reported that Twitter users seemingly disclosed their personal information honestly when they promised to do it, such as *Yata*'s tweet in Figure 2 [10]. As a result, it is easy to collect tweets disclosing submitters' personal profile items when we collect tweets promising to disclose submitters' personal profile items. Furthermore, they often used the same sentence in their tweets, like a game password, as shown in Figure 2, *# I will show the same number of my profile items as your likes*. In order to collect tweets promising to disclose submitters' personal profile items, we used the shared sentence as key to collect them. To be specific, we collected these tweets by using Twitter API v2 [16]. Twitter API v2 helps us to collect tweets where the given sentence is used. Also, Twitter API v2 helps us to collect user accounts who submitted a specific tweet and who gave likes to it. Every 10 PM, we tried to collect user accounts and their tweets

- that contained *# I will show the same number of my profile items as your likes*
- that were submitted in the past 24 hours, and
- that were given one or more likes.

After we obtained the tweets promising to disclose submitters' personal profile items, we tried to collect user accounts who gave likes to the obtained tweets every 10 PM for a week. Finally, we collected 318 Japanese tweets promising to disclose submitters' personal information. These 318 tweets were submitted from December 30, 2021 to January 31, 2022 by 317 users. One out of the 317 users submitted two tweets promising to disclose his personal information on January 12 and 17, 2022. These 318 tweets were given 7060 likes by 6325 users within a week after they were submitted. These 318 tweets can be classified into

- 30 tweets that were deleted during the investigation period (one week), and
- 288 tweets that were not.

The 30 tweets and the other 288 tweets were given 708 and 6352 likes, respectively. Figure 3 shows the histogram of the number of likes given to the 288 tweets that were not deleted during the investigation period, specifically within a week after they were submitted. Figure 4 shows the daily number of likes given to the 288 tweets in the investigation period. Day N in Figure 4 means that N days have passed since the obtained tweet was submitted and our investigation started. Day 6 was the last day of the investigation period. Figure 4 shows that 77 % and 15 % of likes were given on Day 0 and Day 1, respectively. Figure 5 shows the histogram of the number of likes given to the 30 tweets that were deleted during the investigation period (one week). Figure 6 shows the daily number of likes given to the deleted 30 tweets in the investigation period. Figure 6 shows that 80 % and 14 % of likes were given on Day 0 and Day 1, respectively. As shown in Figure 4 and Figure 6, the distribution of the daily number of likes given to the deleted 30 tweets is similar to that of the other 288 tweets. Figure 7 shows the daily number of deleted tweets promising to disclose submitters' personal information in the investigation period. As shown in Figure 7, tweets were most often deleted on Day 4.

IV. An investigation of users concerned in tweets promising to disclose submitters' personal information

In this section, we investigate whether users concerned with a tweet promising to disclose submitter's personal information followed each other. To be specific, we survey

- Twitter users who submitted tweets promising to disclose the same number of their own personal profile items as likes and
- Twitter users who gave likes to these tweets

and investigate

- whether an user who submitted a tweet promising to disclose his/her personal information followed users who gave likes to the tweet,
- whether users who gave likes to a tweet promising to disclose submitter's personal information followed the submitter, and
- whether users who gave likes to a tweet promising to disclose submitter's personal information followed each other.

The investigation is based on an idea: when an user follow someone on Twitter, he/she is not a stranger to the user. We can know whether an user follows someone on Twitter by using Twitter API v2.

After collecting user accounts of submitters and users who gave likes to tweets promising to disclose submitters' personal information, we analyze the relations between them. The relations between a submitter and an user who gave a like to a tweet promising to disclose submitter's personal information can be classified into three types:

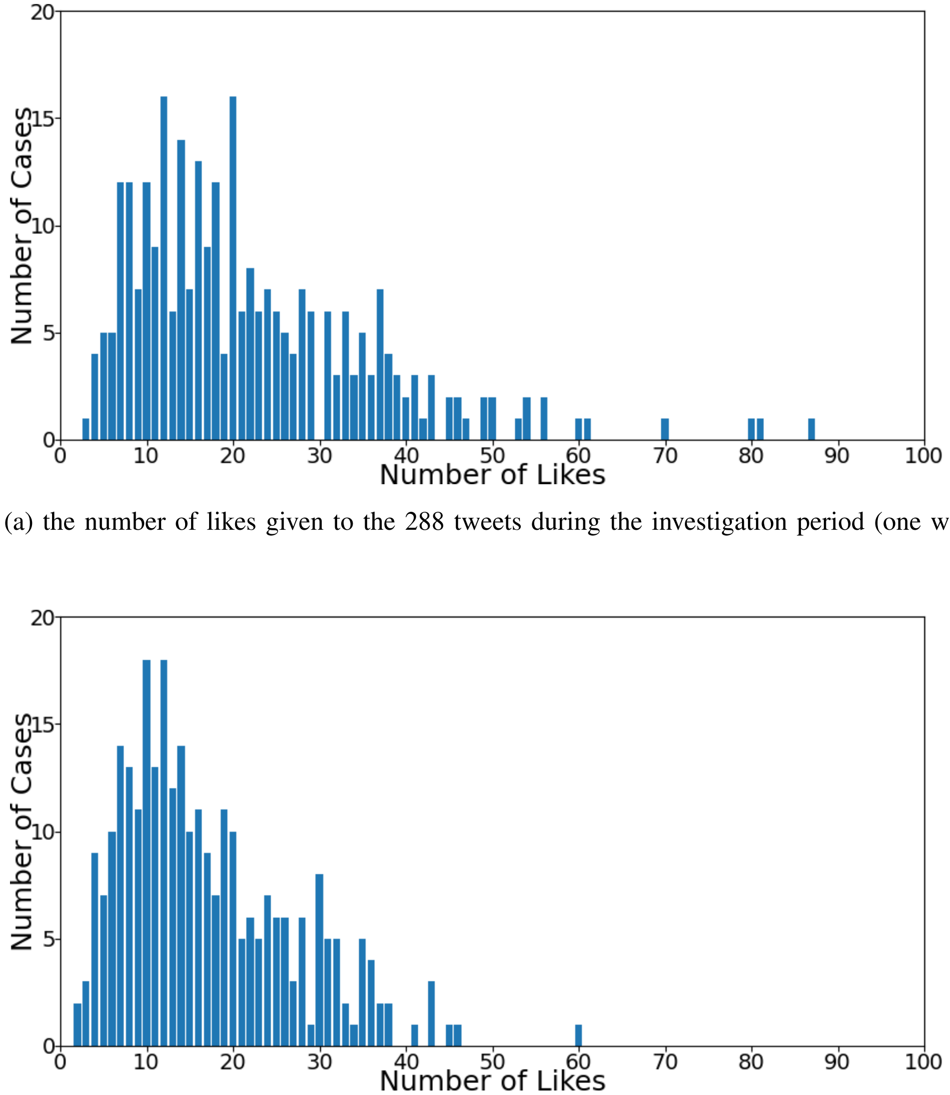
- mutual follow relation: the submitter and the user mutually followed each other.
- one sided follow relation: the submitter followed the user, however, the user did not. Or, the user followed the submitter, however, the submitter did not.
- no follow relation: the submitter and the user did not follow each other.

Furthermore, we analyze the relations among users who gave likes to a tweet promising to disclose submitter's personal information. They can also be classified into three types: mutual follow relation, on sided follow relation, or no follow relation.

Let us consider one example. As shown in Figure 2, a Twitter user, *Yata*, submitted a tweet promising her audiences to disclose the same number of her own personal profile items as likes on December 30, 2021 at 9:02 PM. We detected her tweet on the same day at 10:00 PM, and then, recorded that she received five likes and submitted three replies disclosing her five personal profile items on December 30, 2021. After that, every 10 PM, we tried to check whether someone gave likes to her tweet. On January 5, 2022, we confirmed that five users gave five likes to her tweet on December 30, 2021, as shown in Figure 2, and finished the investigation on her tweet. Then, we analyzed the relations between *Yata* and each of the five users and confirmed that she followed them and each of them followed her. As a result, the relations between *Yata* and each of the five users were mutual follow relations. Furthermore, we analyzed the relations among the five users. There were ten cases to choose two out of the five users. In one case out of the ten, two users followed each other. On the other hand, in nine cases out of the ten, two users did not follow each other. As a result, the relation of one case was a mutual follow relation and the relations of the other nine cases were no follow relations.

As mentioned in Section III, the obtained 318 tweets promising to disclose submitters' personal information can be classified into

- 30 tweets that were deleted during the investigation period (one week), and



(a) the number of likes given to the 288 tweets during the investigation period (one week).

(b) the number of likes given to the 288 tweets on the first day of the investigation period.

Figure 3. The histogram of the number of likes given to the 288 tweets that promised to disclose submitters' personal information and were not deleted during the investigation period (one week).

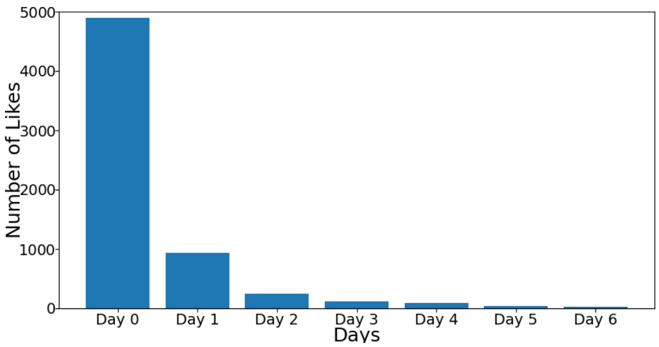


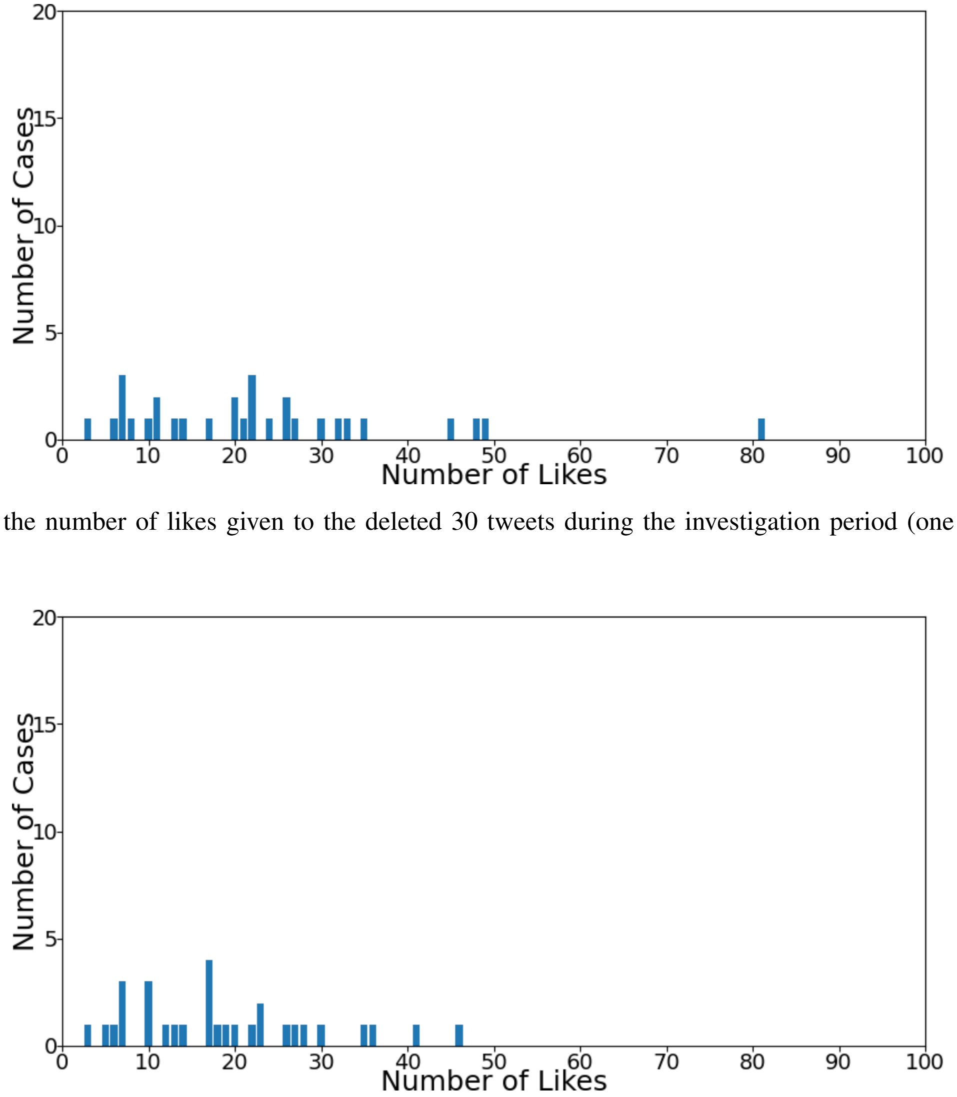
Figure 4. The daily number of likes given to the 288 tweets since the tweets were submitted.

- 288 tweets that were not.

In this section, we survey the 288 tweets that were not deleted during the investigation period and investigate the relations between the submitters of the 288 tweets and the users who gave likes to them.

A. Follow relations between submitters and users who gave likes to tweets promising to disclose submitters' personal information

At first, we discuss the mutual follow relations between submitters and users who gave likes to tweets promising to disclose submitters' personal information. In order to discuss this problem, we introduce the ratio of mutual follow relations between a submitter and users who gave likes to his/her tweet. Suppose that the number of users who gave likes to tweet t is n and m of them are mutually following the submitter of



(a) the number of likes given to the deleted 30 tweets during the investigation period (one week).

(b) the number of likes given to the deleted 30 tweets on the first day of the investigation period.

Figure 5. The histogram of the number of likes given to the deleted 30 tweets promising to disclose submitters' personal information.

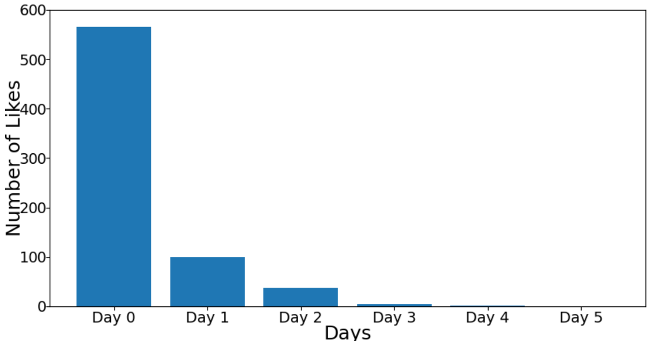


Figure 6. The daily number of likes given to the deleted 30 tweets since the tweets were submitted.

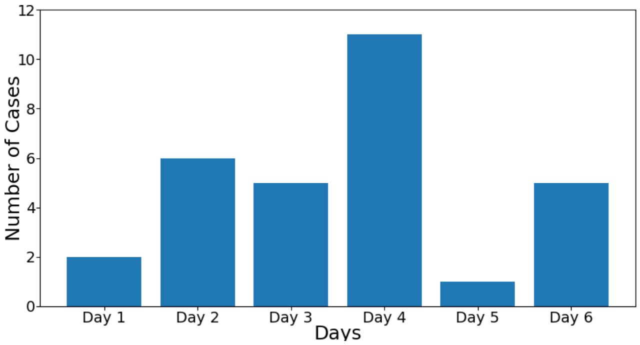
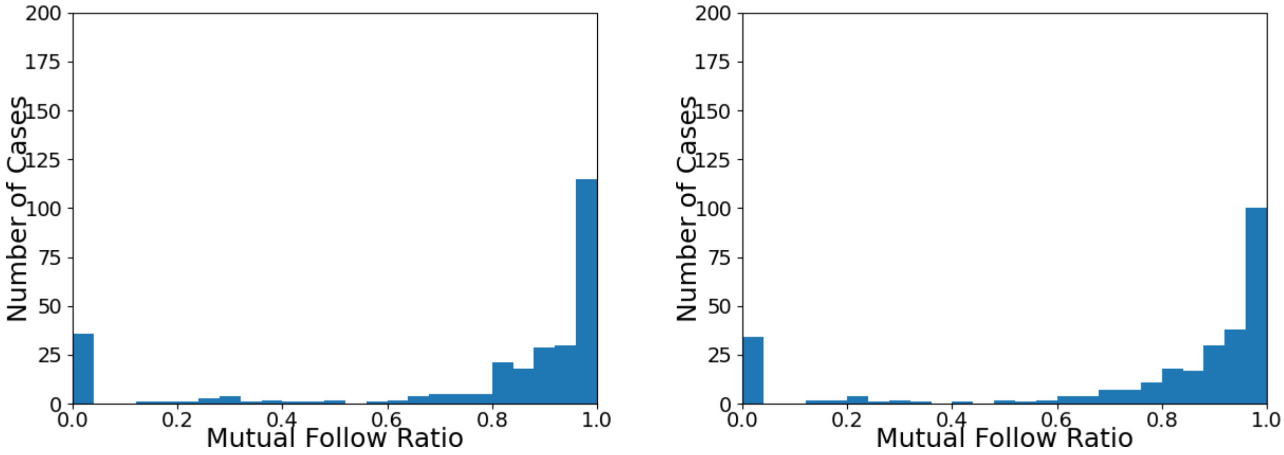


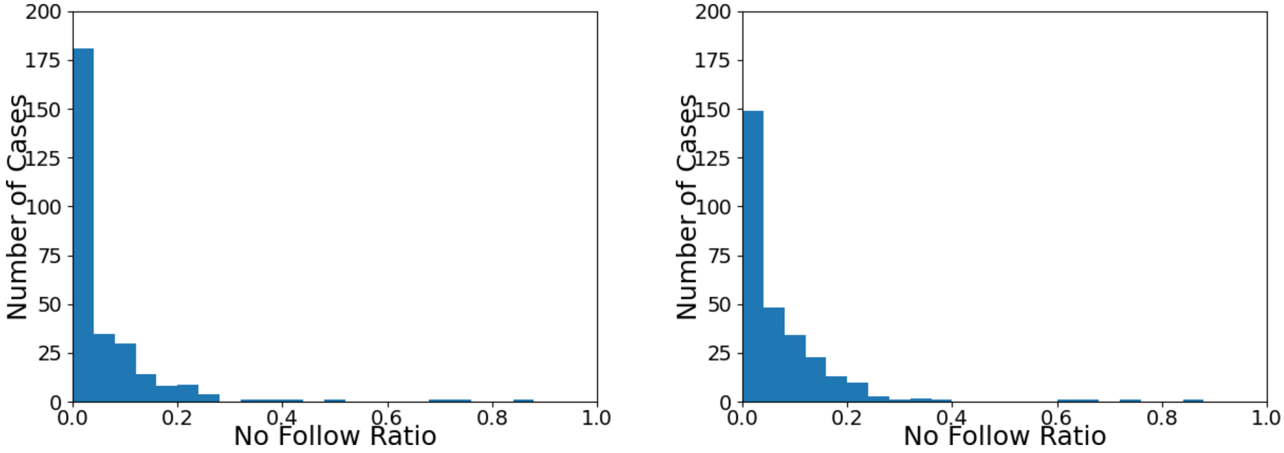
Figure 7. The daily number of deleted tweets promising to disclose submitters' personal information since the tweets were submitted.



(a) the first day (Day 0)

(b) the last day (Day 6)

Figure 8. The histograms of the ratio of mutual follow relations between the submitters of the 288 tweets that were not deleted during the investigation period and the users who gave likes to them on the first day (Day 0) and the last day (Day 6) of the investigation period.



(a) the first day (Day 0)

(b) the last day (Day 6)

Figure 9. The histograms of the ratio of no follow relations between the submitters of the 288 tweets that were not deleted during the investigation period and the users who gave likes to them on the first day (Day 0) and the last day (Day 6) of the investigation period.

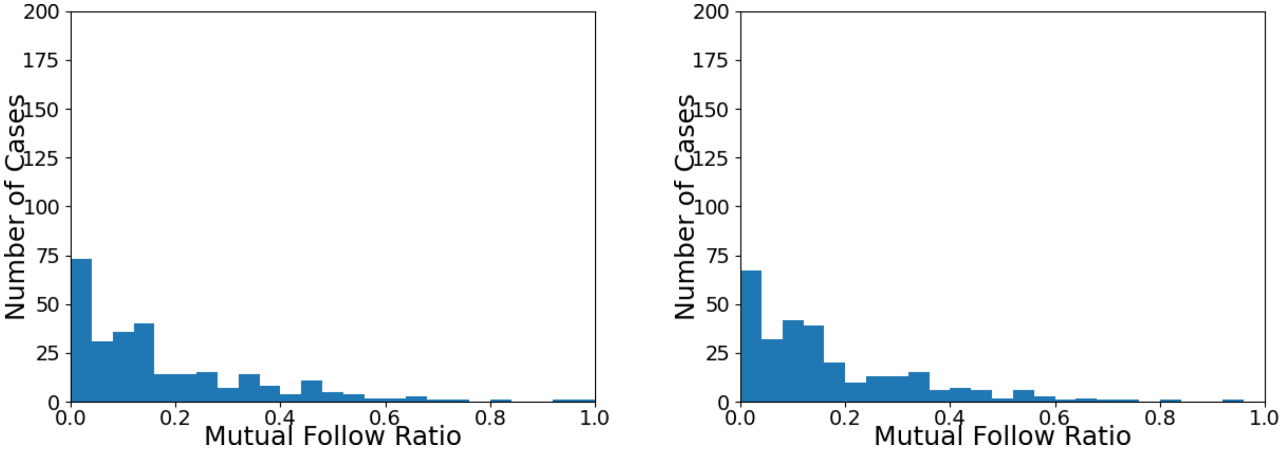
tweet t. Then, the ratio of mutual follow relations between the submitter of tweet t and the users who gave likes to it, $P_{MF1}(t)$, is defined as follows:

$$P_{MF1}(t) = \frac{m}{n}$$

Figure 8 shows the distribution of the ratio of mutual follow relations between the submitters of the 288 tweets that were not deleted during the investigation period and the users who gave likes to them. Furthermore, Figures 8 (a) and (b) show the distribution of them investigated on the Day 0 and Day 6, respectively. As shown in Figure 8, it is probable that most of the users have followed the submitters mutually before they gave likes to submitters' tweets promising to disclose their personal information. In other words, the submitters and most of the users were not strangers to each other. The distribution of the mutual relation ratio on Day 6 (Figure 8 (b)) moved to the left than that on Day 0 (Figure 8 (a)). It showed that the number of users who did not follow the submitters and whom the submitters did not follow increased. It is probable that submitters were careful to follow unfamiliar users even if they gave likes to their tweets.

Next, we discuss the no follow relations between submitters and users who gave likes to tweets promising to disclose submitters' personal information. In order to discuss this problem, we introduce the ratio of no follow relations between a submitter and users who gave likes to his/her tweet. Suppose that the number of users who gave likes to tweet t is n and l of them are not following the submitter of tweet t and the submitter is not following them, too. Then, the ratio of no follow relations between the submitter of tweet t and the users who gave likes to it, $P_{NF1}(t)$, is defined as follows:

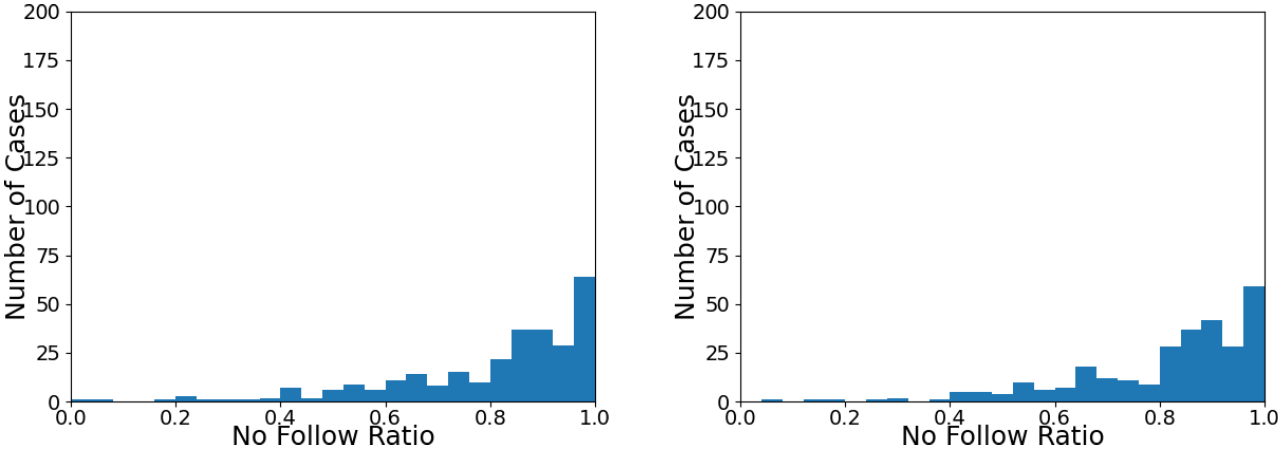
$$P_{NF1}(t) = \frac{l}{n}$$



(a) the first day (Day 0)

(b) the last day (Day 6)

Figure 10. The histograms of the ratio of mutual follow relations among the users who gave likes to the 288 tweets that were not deleted during the investigation period on the first day (Day 0) and the last day (Day 6) of the investigation period.



(a) the first day (Day 0)

(b) the last day (Day 6)

Figure 11. The histograms of the ratio of no follow relations among the users who gave likes to the 288 tweets that were not deleted during the investigation period on the first day (Day 0) and the last day (Day 6) of the investigation period.

Figure 9 shows the distribution of the ratio of no follow relations between the submitters of the 288 tweets that were not deleted during the investigation period and the users who gave likes to them. Figure 9 shows that the number of users who had the no follow relations with the submitters was small on Day 0 and increased since then. It is probable that the delays were caused by the time it took to find tweets disclosing submitters' personal information.

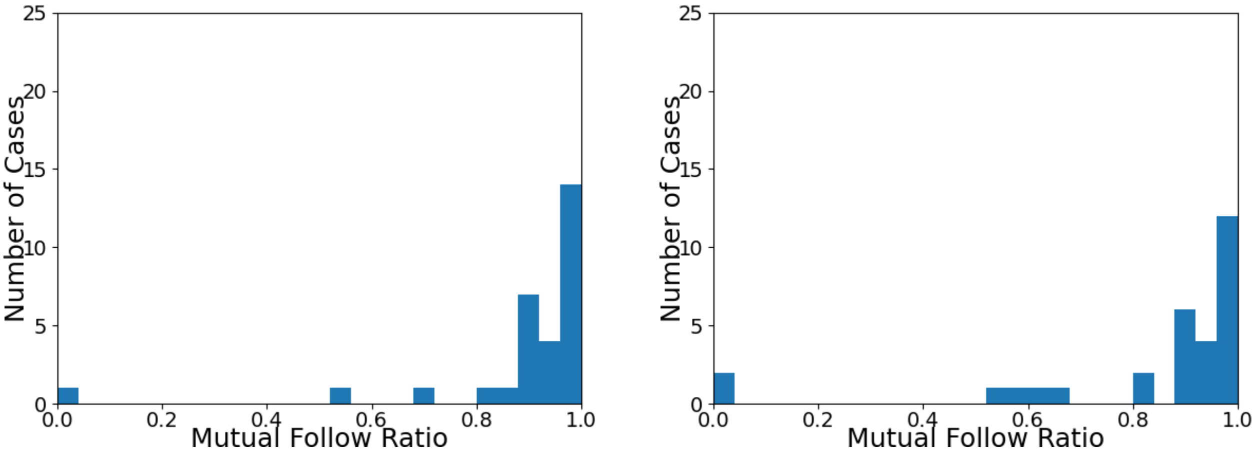
B. Follow relations among users who gave likes to tweets promising to disclose submitters' personal information

At first, we discuss the mutual follow relations among users who gave likes to tweets promising to disclose submitters' personal information. In order to discuss this problem, we introduce the ratio of mutual follow relations among users who gave likes to a tweet. Suppose that the number of users who gave likes to tweet t is n and there are m cases where two users of them are following each other. Then, the ratio of mutual follow relations among the users who gave likes to tweet t, $P_{MF2}(t)$, is defined as follows:

$$P_{MF2}(t) = \frac{m}{n(n-1)/2}$$

Figure 10 shows the distribution of the ratio of mutual follow relations among the users who gave likes to the 288 tweets that were not deleted during the investigation period. Figure 10 shows that it is probable that most of the users did not follow each other mutually. In other words, most of the users were strangers to each other although they followed the same submitters and gave likes to their tweets.

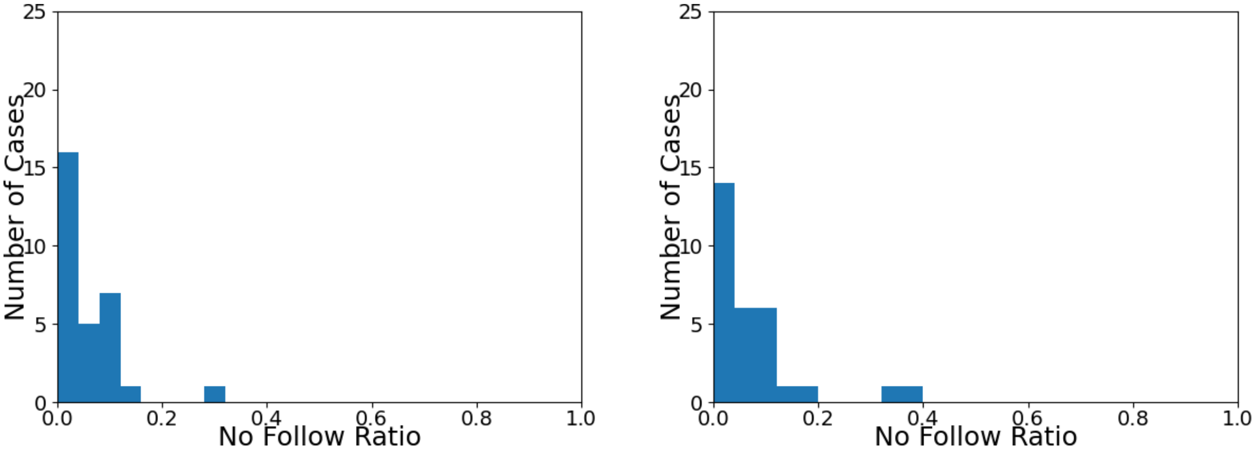
Next, we discuss the no follow relations among users who gave likes to tweets promising to disclose submitters' personal information. In order to discuss this problem, we introduce the ratio of no follow relations among users who gave likes to a



(a) the first day (Day 0)

(b) one day prior to the day they were deleted

Figure 12. The histograms of the ratio of mutual follow relations between the submitters of the deleted 30 tweets and the users who gave likes to them on the first day (Day 0) and one day prior to the day they were deleted.



(a) the first day (Day 0)

(b) one day prior to the day they were deleted

Figure 13. The histograms of the ratio of no follow relations between the submitters of the deleted 30 tweets and the users who gave likes to them on the first day (Day 0) and one day prior to the day they were deleted.

tweet. Suppose that the number of users who gave likes to tweet t is n and there are l cases where two users of them are not following each other. Then, the ratio of no follow relations among the users who gave likes to tweet t, $P_{NF2}(t)$, is defined as follows:

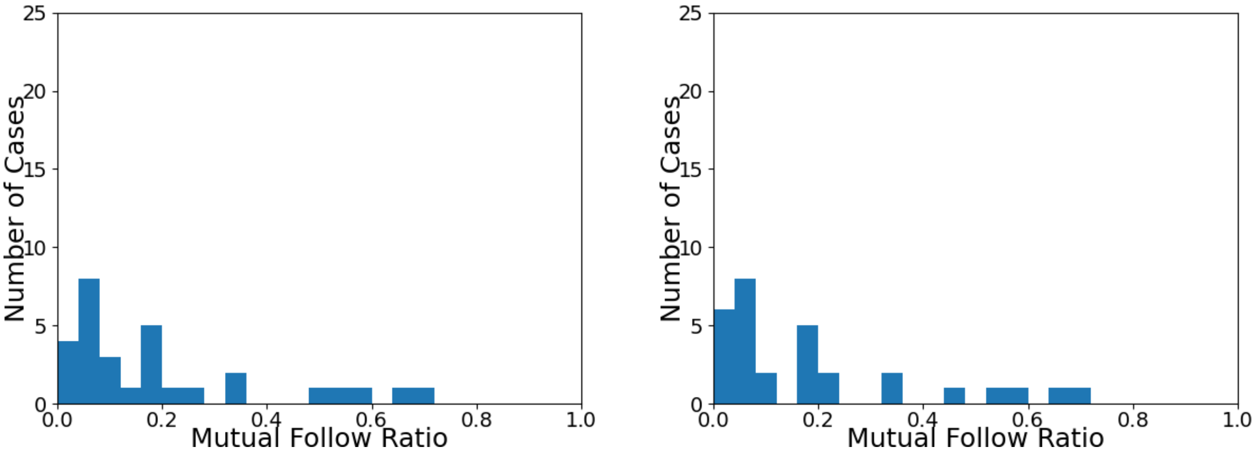
$$P_{NF2}(t) = \frac{l}{n(n-1)/2}$$

Figure 11 shows the distribution of the ratio of no follow relations among the users who gave likes to the 288 tweets that were not deleted during the investigation period. The distribution of the no relation ratio on Day 6 (Figure 11 (b)) was similar to that on Day 0 (Figure 11 (a)). It showed that it is probable that not many users started to follow users within a week even if they gave likes to the same tweets. It is probable that users were careful to follow unfamiliar users even if they gave likes to the same tweets.

V. An Investigation of Users Concerned in Deleted Tweets Promising to Disclose Submitters' Personal Information

As mentioned in Section III, the obtained 318 tweets promising to disclose submitters' personal information contained 30 tweets that were deleted during the investigation period, specifically within a week after they were submitted. In this section, we investigate the relations between the submitters of the deleted 30 tweets and the users who gave likes to them.

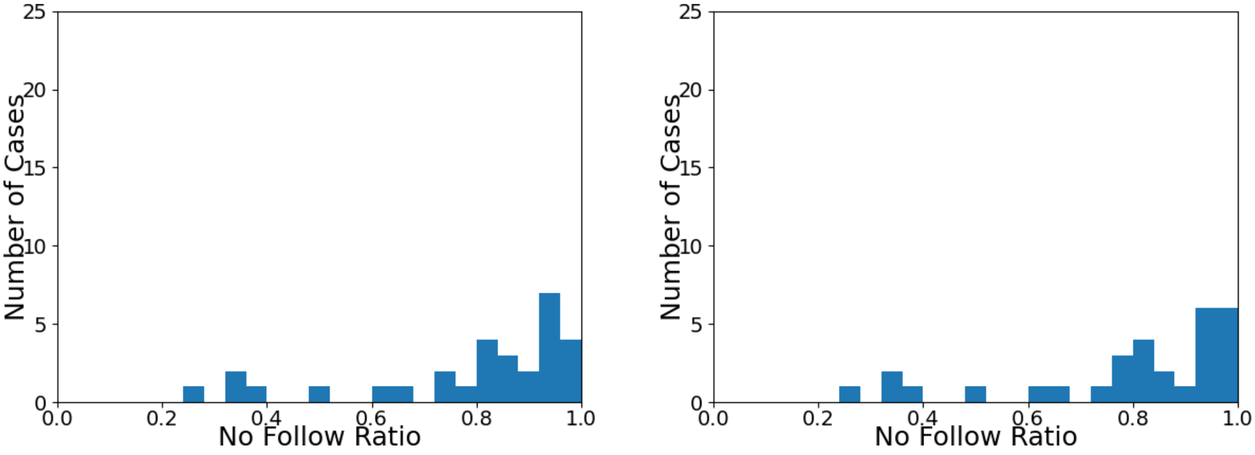
Figure 12 shows the distribution of the ratio of mutual follow relations between the submitters of the deleted 30 tweets and the users who gave likes to them. Furthermore, Figures 12 (a) and (b) show the distribution of them investigated on the Day 0 and one day prior to the day they were deleted, respectively. As in Figure 8, Figure 12 shows that it is probable that most of the users have followed the submitters mutually before they gave likes to submitters' tweets promising



(a) the first day (Day 0)

(b) one day prior to the day they were deleted

Figure 14. The histograms of the ratio of mutual follow relations among the users who gave likes to the deleted 30 tweets on the first day (Day 0) and one day prior to the day they were deleted.



(a) the first day (Day 0)

(b) one day prior to the day they were deleted

Figure 15. The histograms of the ratio of no follow relations among the users who gave likes to the deleted 30 tweets on the first day (Day 0) and one day prior to the day they were deleted.

to disclose their personal information. In other words, the submitters and most of the users were not strangers to each other for some time.

Figure 13 shows the distribution of the ratio of no follow relations between the submitters of the deleted 30 tweets and the users who gave likes to them. As in Figure 9, Figure 13 shows that the number of users who had the no follow relations with the submitters was small.

Figure 14 and Figure 15 show the distribution of the ratio of mutual follow relations and no follow relations among the users who gave likes to the deleted 30 tweets, respectively. As in Figure 10 and Figure 11, Figure 14 and Figure 15 show that it is probable that many of the users did not follow each other mutually although they followed the same submitters and gave likes to their tweets. In other words, many of the users were strangers to each other even if they followed the same submitters and gave likes to the same tweets.

In this way, the follow relations of users concerned with the deleted 30 tweets promising to disclose submitters' personal information are similar to those of users concerned with the 288 tweets that were not deleted during the investigation period. As a result, the results of our survey showed that, whether or not submitters deleted their tweets disclosing their personal information in short order,

- the submitters followed most of the users mutually before the users gave likes to tweets promising to disclose submitters' personal information, and
- most of the users did not follow each other although they followed the same submitters and gave likes to their tweets.

VI. Conclusion

In this paper, we investigated the relations of submitters of tweets promising to disclose their personal information and users who gave likes to the tweets, taking into account whether or not the submitters deleted their tweets in short order. The results of our investigation show that most of the users had followed the submitters mutually before they gave likes to submitters' tweets promising to disclose their personal information whether or not the submitter deleted their tweets in short order. On the other hand, most of the users did not follow each other although they followed the same submitters and gave likes to their tweets. As time went on, the number of users who gave likes to submitters' tweets promising to disclose their personal information but did not follow the submitters and whom the submitters did not follow increased. It is probable that submitters were careful to follow unfamiliar users even if they gave likes to their tweets. Also, users were careful to follow unfamiliar users even if they followed the same submitters and gave likes to the same tweets. The system that understands these relations might carefully treat users who choose not to friend someone with good reasons.

References

[1] Y. Watanabe, T. Nakano, H. Nishimura, and Y. Okada, "An Investigation of Twitter Users Who Gave Likes to Tweets Disclosing Submitters' Personal Information," in Proceedings of the Eighth International Conference on Human and Social Analytics (HUSO 2022), May 2022, pp. 10–15. [Online]. Available: https://www.thinkmind.org/index.php?view=article&articleid=huso_2022_1_30_80026 [accessed: 2022-12-12]

[2] C. Johnson III, Safeguarding against and responding to the breach of personally identifiable information, Office of Management and Budget Memorandum, 2007. [Online]. Available: https://obamawhitehouse.archives.gov/sites/default/files/omb/assets/omb/memoranda/fy2007/m07-16.pdf [accessed: 2022-12-12]

[3] B. Krishnamurthy and C. E. Wills, "On the leakage of personally identifiable information via online social networks," Computer Communication Review, vol. 40, no. 1, 2010, pp. 112–117. [Online]. Available: https://doi.org/10.1145/1672308.1672328 [accessed: 2022-12-12]

[4] S. Fox et al., Trust and Privacy Online: Why Americans Want to Rewrite the Rules, The Pew Internet & American Life Project, 2000. [Online]. Available: http://www.pewinternet.org/2000/08/20/trust-and-privacy-online/ [accessed: 2022-12-12]

[5] A. Acquisti and R. Gross, "Imagined Communities: Awareness, Information Sharing, and Privacy on the Facebook," in Proceedings of the 6th International Conference on Privacy Enhancing Technologies, ser. PET'06. Berlin, Heidelberg: Springer-Verlag, 2006, pp. 36–58. [Online]. Available: https://doi.org/10.1007/11957454_3 [accessed: 2022-12-12]

[6] S. Livingstone, "Taking risky opportunities in youthful content creation: teenagers' use of social networking sites for intimacy, privacy and self-expression." New Media & Society, vol. 10, no. 3, 2008, pp. 393–411. [Online]. Available: https://journals.sagepub.com/doi/10.1177/1461444808089415 [accessed: 2022-12-12]

[7] S. B. Barnes, "A privacy paradox: Social networking in the United States." First Monday, vol. 11, no. 9, 2006. [Online]. Available: http://firstmonday.org/article/view/1394/1312 [accessed: 2022-12-12]

[8] A. Viseu, A. Clement, and J. Aspinall, "Situating privacy online: Complex perception and everyday practices," Information, Communication & Society, 2004, pp. 92–114. [Online]. Available: https://doi.org/10.1080/1369118042000208924 [accessed: 2022-12-12]

[9] T. Hirai, "Why does "Enjyo" happen on the Web? : An Examination based on Japanese Web Culture," Journal of Information and Communication Research, vol. 29, no. 4, mar 2012, pp. 61–71. [Online]. Available: http://doi.org/10.11430/jsicr.29.4_61 [accessed: 2022-12-12]

[10] Y. Watanabe, H. Nishimura, Y. Chikuki, K. Nakajima, and Y. Okada, "An Investigation of Twitter Users Who Disclosed Their Personal Profile Items in Their Tweets Honestly," in Proceedings of the Sixth International Conference on Human and Social Analytics (HUSO 2020), Oct 2020, pp. 20–25. [Online]. Available: http://www.thinkmind.org/index.php?view=article&articleid=huso_2020_1_40_80035 [accessed: 2022-12-12]

[11] M. S. Granovetter, "The Strength of Weak Ties," American Journal of Sociology, vol. 78, no. 6, 1973, pp. 1360–1380.

[12] C. Haythornthwaite, "Strong, Weak and Latent Ties and the Impact of New Media," The Information Society, vol. 18, no. 5, October 2002, pp. 385–401. [Online]. Available: https://doi.org/10.1080/01972240290108195 [accessed: 2022-12-12]

[13] N. B. Ellison, J. Vitak, R. Gray, and C. Lampe, "Cultivating Social Resources on Social Network Sites: Facebook Relationship Maintenance Behaviors and Their Role in Social Capital Processes," Journal of Computer-Mediated Communication, vol. 19, no. 4, 07 2014, pp. 855–870. [Online]. Available: https://doi.org/10.1111/jcc4.12078 [accessed: 2022-12-12]

[14] K. Panovich, R. Miller, and D. Karger, "Tie Strength in Question & Answer on Social Network Sites," in Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work, ser. CSCW '12. New York, NY, USA: Association for Computing Machinery, 2012, p. 10571066. [Online]. Available: https://doi.org/10.1145/2145204.2145361 [accessed: 2022-12-12]

[15] E. Gilbert and K. Karahalios, "Predicting Tie Strength with Social Media," in Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, ser. CHI '09. New York, NY, USA: Association for Computing Machinery, 2009, p. 211220. [Online]. Available: https://doi.org/10.1145/1518701.1518736 [accessed: 2022-12-12]

[16] Twitter, Inc. Twitter API. [Online]. Available: https://developer.twitter.com/en/docs/twitter-api [accessed: 2022-12-12]