

Article

The Concept of Interactive Dynamic Intelligent Virtual Sensors (IDIVS): Bridging the Gap between Sensors, Services, and Users through Machine Learning

Jan A. Persson ^{*}, Joseph Bugeja , Paul Davidsson , Johan Holmberg , Victor R. Kebande , Radu-Casian Mihailescu, Arezoo Sarkheyli-Hägele and Agnes Tegen

Internet of Things and People Research Center, Department of Computer Science and Media Technology, Malmö University, 205 06 Malmö, Sweden

* Correspondence: jan.a.persson@mau.se

Abstract: This paper concerns the novel concept of an Interactive Dynamic Intelligent Virtual Sensor (IDIVS), which extends virtual/soft sensors towards making use of user input through interactive learning (IML) and transfer learning. In research, many studies can be found on using machine learning in this domain, but not much on using IML. This paper contributes by highlighting how this can be done and the associated positive potential effects and challenges. An IDIVS provides a sensor-like output and achieves the output through the data fusion of sensor values or from the output values of other IDIVSs. We focus on settings where people are present in different roles: from basic service users in the environment being sensed to interactive service users supporting the learning of the IDIVS, as well as configurators of the IDIVS and explicit IDIVS teachers. The IDIVS aims at managing situations where sensors may disappear and reappear and be of heterogeneous types. We refer to and recap the major findings from related experiments and validation in complementing work. Further, we point at several application areas: smart building, smart mobility, smart learning, and smart health. The information properties and capabilities needed in the IDIVS, with extensions towards information security, are introduced and discussed.

Keywords: virtual sensors; soft sensors; sensor data fusion; online learning; interactive machine learning; transfer learning; information security



Citation: Persson, J.A.; Bugeja, J.; Davidsson, P.; Holmberg, J.; Kebande, V.R.; Mihailescu, R.-C.; Sarkheyli-Hägele, A.; Tegen, A. The Concept of Interactive Dynamic Intelligent Virtual Sensors (IDIVS): Bridging the Gap between Sensors, Services, and Users through Machine Learning. *Appl. Sci.* **2023**, *13*, 6516. <https://doi.org/10.3390/app13116516>

Academic Editor: Antonio Fernández-Caballero

Received: 22 December 2022

Revised: 13 March 2023

Accepted: 24 May 2023

Published: 26 May 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The development of sensor technology and the Internet of Things (IoT) has opened up new opportunities based on the increasing amount of data being generated and made available. Many of the applications we foresee rely on a good estimation of the current state of an environment. Examples of such application areas are smart buildings, smart cities, smart mobility, and smart health. Sometimes the relevant aspect of an environment's state is not directly detectable by using available physical sensors, e.g., the detection of what activity is taking place in a room. Hence, several different sensor values may need to be fused in order to estimate the current state. We introduce the Interactive Dynamic Intelligent Virtual Sensor (IDIVS), which, in particular, capitalizes on interactive machine learning to make use of people being present in the environment. The IDIVS is a logical entity that takes sensor data or data from other IDIVSs as input and produces novel data about the current state of an environment (numerical or categorical) in real time. It is based on previous work in the area about a DIVS (Dynamic Intelligent Virtual Sensor) [1,2]. The idea is that the model for data fusion is updated by interactive machine learning to produce a more accurate output. In particular, the IDIVS can make use of feedback from people that are able to both observe the output of the IDIVS and the actual state of the environment, and thereby might be triggered to provide correct labels or even suggest new state classes. By making use of online learning in this fashion, new challenges arise with respect to

information security, because new types of attacks on the system can be launched, e.g., by having people manipulate the classification scheme to fit their own agenda. For the IDIVS to produce as much accuracy as possible and as quickly as possible after being installed, we also explore transfer learning and user interaction interfaces, which may be particularly relevant in cold start situations.

By using these different learning mechanisms, an IDIVS is also able to handle situations where the set of sensors streaming data is not fixed over time, but dynamic. Sensors may malfunction or lose network connectivity and stop providing sensor data. Sensor data may come from portable devices, such as smartphones, which only appear and provide sensor data temporarily. Further, dedicated stationary devices with sensors may be installed in or removed from the environment permanently. The environment may also pose challenges, e.g., sensors and physical objects, such as chairs, may be moved. Hence, it is important that the IDIVS is adaptable. An important mechanism for the IDIVS to cope with the dynamicity of the sensors and the environment is again the online machine learning capability, primarily based on the fact that there are humans present that can provide feedback.

The vision of the IDIVS is that it can constitute a piece of software that can be re-used as a building block where a service needs information on the current state or context. The output of the IDIVS may be used by different applications/services. The concept creates opportunities to better meet the local/contextual needs of providing services in different settings (buildings, transport, health applications, etc.). Hence, we argue it constitutes the missing link between streamed sensor data from multiple systems and the services/applications layers to enable new or improved value-added services.

The purpose of this paper is to further expand the earlier suggested DIVS [1,2] concept towards being more refined concerning aspects connected to interactive machine learning and transfer learning, and to highlight the information security aspects that come into play due to the learning. The paper will also point to some recent related work that validates a number of properties of the suggested concept. Hence, the paper contributes by clarifying how interactive machine learning has the potential to improve state detection in a number of application domains by extending the concept of virtual sensors. Further, it highlights the challenges that come with this extension. Finally, it contributes by exemplifying how methods such as transfer learning can additionally strengthen the concept.

The remainder of the paper introduces related work. Thereafter, the IDIVS concept is introduced and specified. Application areas for the IDIVS are exemplified. Next, the aspects of interactive machine learning and transfer learning in IDIVS are discussed, and a number of results are presented. Since the IDIVS concept brings in human interaction, a potentially new attack vector on the system is discussed. A section discusses information security concerns and possible mitigations. The paper culminates with several conclusions and pointers to future research.

Related Work and Research Gap

Situation/activity recognition is an important task in understanding the context of a situation in order to achieve services in different domains. Virtualization is viewed as a key aspect of the IoT to support multiple applications and services efficiently [3]. Virtualization of sensors is not particularly new (see, for instance, early work by Liu et al. [4] and the survey in Islam et al. [5]). Early work included feedback loops via Kalman filtering for improving the sensing ability and providing efficient active noise cancelation. Design aspects and requirements of virtualization [6], real-time formulation of virtual sensors [7], and re-configuration [8] have been dealt with. Little attention, however, has been placed on opportunities attainable with humans in the loop, including regular users/non-experts involved in interactive learning in the context of virtual sensors. Interactive learning includes both active learning [9], where the machine requests labels for data instances, and machine teaching, where the human proactively provides suitable training data [10]. Both of these concepts and combinations of them are considered relevant for the IDIVS concept.

Machine learning techniques can be applied to deal with the information provided by sensors and virtual sensors in this task. The idea of utilizing machine learning for data fusion into soft sensors/virtual sensors has been around for some time (see, for instance, the 2009 paper by Kadlec and Gabrys [11], although focused on process industry applications). Note that we focus on data-driven approaches—as opposed to model-driven—as we strive to use the same or similar IDIVS in multiple application areas. Although the focus of IDIVS is on using supervised learning (the interactive machine learning approach), it does not exclude the complementing use of unsupervised learning approaches. In recent years, researchers have proposed different learning techniques for activity recognition based on virtual constant sensors that are always available (see, for example, Aguilera et al. [12] and the associated review [13]). It is noticeable in many applications of virtual sensors, such as sensors in smartphones, that the set of sensors is constant and is normally assumed to always be available. However, applications of dynamic virtual sensors can potentially produce inconsistent outputs, which is an important concern. The idea of IDIVS is to deal with that concern by applying machine learning techniques to the data fusion process.

As mentioned, there is not much research on using interactive machine learning in connection with virtual sensors in realistic scenarios where the dynamicity of the sensors is considered. This is particularly true if considering virtual sensors and connected approaches that are general enough to be applicable to multiple domains. This view is also strengthened by looking at reviews of IoT and ML approaches (see, for instance, a recent review by Arikumar K.S. et al. [14], which identifies the common practice of using sensor data for ML, but does not identify research leveraging user input, such as used in interactive machine learning).

2. The IDIVS Concept and Application Domains

2.1. The IDIVS Concept

The IDIVS is a logical entity/instance (software) producing a sensor-like output in real time that can be a numerical or categorical value, e.g., the number of people or activity in an environment. IDIVSs may be used in a hierarchical architecture, where an IDIVS may use the output from other IDIVSs as input, some sensors may be hidden from some IDIVSs, and services may access all or a sub-set of IDIVSs' output, as illustrated in Figure 1.

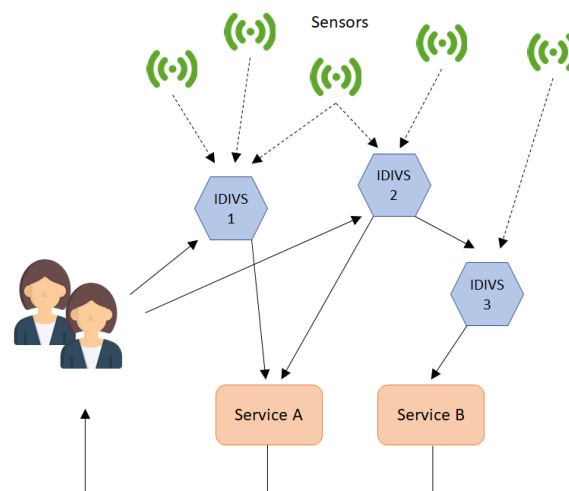


Figure 1. The hierarchical structure of IDIVS and service support.

We assume there are mechanisms for both securing data access, i.e., access control for entities, as well as for supporting the discovery of sensors, IDIVSs, and services. The IDIVSs are primarily, but not exclusively, intended for use in environments where people are or may be present, where they may use services relying on information from the IDIVS, and where they may provide information to the IDIVS, e.g., in interactive learning. We see the following roles a human user may have in relation to the IDIVS.

User roles:

- Service user, i.e., a user of a service relying on the IDIVS output;
- Interactive service user, i.e., a user that also interacts with the system, providing input to the system that can help the system learn/adapt.

IDIVS configuration roles:

- System configurator, i.e., defining what set of sensor/IDIVS to use as input, what states to detect, where to distribute the output, etc.;
- IDIVS teacher, e.g., orchestrating scenarios, which is used for training IDIVSs in a systematic manner.

The idea is that each IDIVS is connected to an environment, e.g., a room in a building or a square in a city, etc., where sensors of different types may be present at different times. It may also be connected to an object or even a person, e.g., being a personal IDIVS.

2.2. Capabilities

An IDIVS performs data fusion to produce an output. Data fusion aims to integrate and aggregate data from multiple sensors to generate more accurate and complete information than would be possible when the sensors were used individually. The IDIVS may incorporate different data fusion types:

- Redundant/competitive fusion: multiple sensor data (of the same type) representing the same information area are integrated to increase the accuracy of the information about the state of an environment;
- Cooperative fusion: multiple sensor data (of different types) representing the same information area are integrated to obtain more information about the state of an environment;
- Complementary fusion: multiple sensor data representing different environments are integrated to obtain complete information about a state. This information is usually provided by multiple IDIVSs connecting to different environments.

The relevant capabilities of IDIVS are to:

- Fuse sensor data to produce a sensor-like output (numerical or categorical) about the state of a specific environment based on sensor-like input (numerical or categorical) relevant for the environment;
- Adapt to changes in the set of sensors, including integrating new relevant sensor values at run-time;
- Use labelled data to improve accuracy by machine learning, e.g., based on user feedback relevant for the current IDIVS output, a batch of pre-labelled data, or a model or partial model of IDIVS (e.g., by transfer learning);
- Support self-assessment of the accuracy of its output;
- Learn to detect new states of its environment (not specified at design time);
- Provide information about the sensor input it uses with respect to how important different sensor values are for generating the output (e.g., a type of information gain);
- Anomaly detection and diagnosis, e.g., detecting tampering attempts on input sensor streams.

2.3. IDIVS Input and Output

The suggested minimal output produced and streamed by the IDIVS is assumed to be an ID and a virtual sensor value (numerical or categorical—holding information about an environment or object).

Additional properties, mandatory or optional for the IDIVS, are shown in Figure 2. It also shows the input retrieved from sensors: ID, sensor value, and associated properties. Additionally, it may be useful that an IDIVS can set the timestamp of a virtual sensor value. Sensor data can be produced and consumed by the IDIVS regularly, irregularly, or on request, synchronously or asynchronously. Obviously, for the IDIVS to make use of machine learning based on user input in a supervised manner, it needs labelled data or equivalent input. The ML problem for this data fusion problem becomes to train the model,

e.g., update the data fusion model such that the virtual sensor output value matches the user input, e.g., the correct labels.

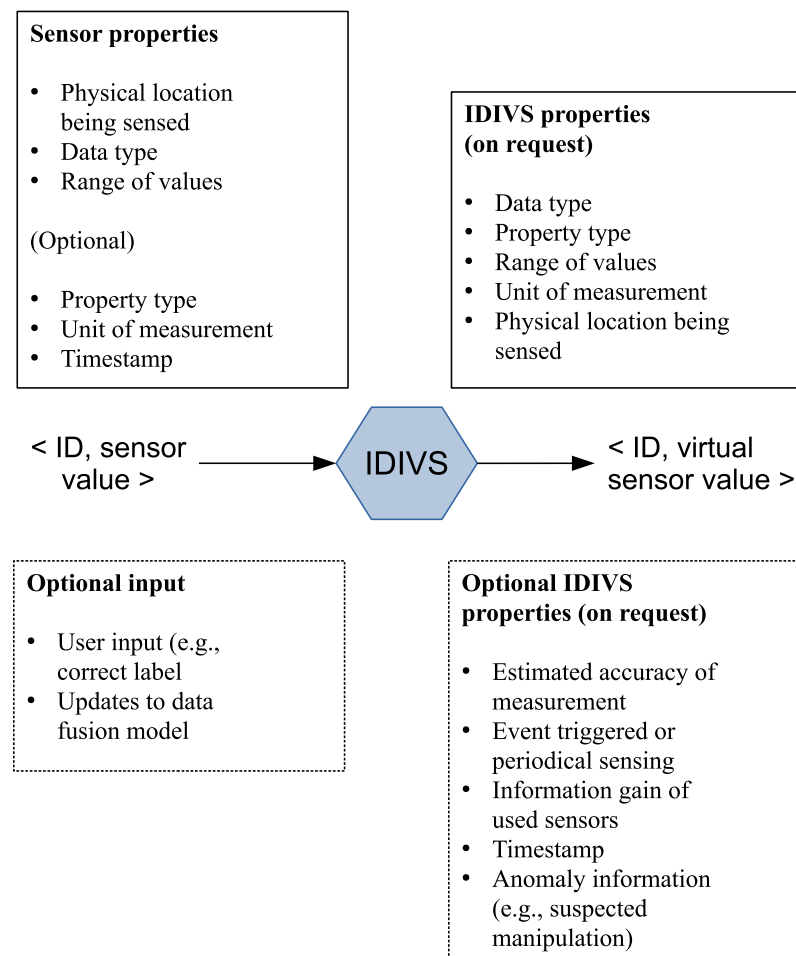


Figure 2. The information properties and input/output of IDIVS.

2.4. Application Areas

Given the output IDIVS can create, different services can be designed. As mentioned above, the output of the IDIVS is a virtual sensor value, typically representing a property of the environment. It can be a physical property as well as other states of the environment, e.g., related to human activities.

Hence, services can be built using IDIVS output (single or multiple) as well as other data sources. A desired property of the service is that it supports the provision of feedback to the IDIVS, enabling learning to improve the accuracy of the IDIVS. Ideally, the feedback constitutes correctly labelled data provided by users of the service. If users provide feedback to improve accuracy, the service becomes interactive to a certain degree. Potentially, this opens up the door to new and richer services and to new areas of application. Next, we exemplify a few application areas. Obviously, IDIVS may be relevant for numerous applications where IoT is used or could be used (see, for instance, reviews of IoT and applications [15]).

2.4.1. Smart Buildings

An application in smart buildings is that an IDIVS is used to detect human activities, e.g., meeting type, presentation, discussions, and/or work type, silent work, phone, or video calls. Other IDIVSs can detect related states, such as the number of people per room or per square metre. The idea is that the user can provide feedback on the activities (either existing or new). In other applications, an IDIVS can be used to detect complex

physical properties, such as air quality (including temperature and odour). The sensors may be both stationary sensors, e.g., motion sensors, sound level detectors, image/camera sensors, as well as mobile sensors in smartphones or on movable furniture. Potential applications (services) exist, for instance, connected to building and equipment automation, e.g., controlling blinds and presentation equipment, HVAC control adapted to the number of people present; guidance of people, e.g., finding a suitable room for the work at hand; and gathering usage data of a building, e.g., for re-allocating the types of rooms in a building or maintenance. Other examples of potential applications in smart buildings can be found in the review by Jia et al. [16].

2.4.2. Smart Mobility

To estimate the CO₂ footprint of an individual caused by urban transportation, we need to know which means of transportation a person is currently using. To estimate the current mode of travel, an IDIVS could be integrated into a person's smartphone using, e.g., position and accelerator data as input data, and leverage user feedback to improve its accuracy in estimating the current mode. Another example of an output of an IDIVS in smart mobility is the number of people in a vehicle, e.g., in a bus or a train, based on multiple sensors, e.g., door entries and Wi-Fi units connected. This, in turn, can be used for, e.g., guiding travellers to a specific bus departure to avoid crowding. In this case, drivers of vehicles can occasionally provide the correct count of passengers to the IDIVS. In general, there are ample examples of IoT-based services in the domain [17,18] that could also be relevant for the IDIVS concept.

2.4.3. Smart Health

The IDIVS can be used to estimate and output the different activities humans perform, e.g., sitting still, resting, walking, biking, etc., which can be used in different health applications. Self-reflection on personal data has been shown to have a positive impact on the overall quality of a user's health. Whether the purpose is to monitor one's own health and well-being or to change habits for a healthier lifestyle, the IDIVS can potentially enhance the capabilities of personal sensors, such as smartphones, smart watches, and wearable technology, as well as make use of stationary sensors, e.g., motion sensors, to estimate richer and more elaborated states. It can also be applied, for instance, to work ergonomics by tracking a user's activities. In the examples above, the user feedback makes the system more interactive and accurate. Furthermore, the active engagement of the user may make them reflect on their activities during the day. For recent examples of the IoT in the healthcare domain, see the review by Chen et al. [19].

2.4.4. Smart Learning

One of the issues with remote learning or digital learning is the lack of adequate interactivity that often happens in other learning settings. In particular, it is a challenge for the teacher to obtain responses on the state of the student, e.g., concerning the student's concentration level or the need for a break. IDIVSs may be used for estimating and outputting such states of students by, e.g., monitoring keyboard activities or by face image processing. The feedback from students can be direct about the correctness of assessed states as well as based on various learning results, e.g., from quizzes. This can then be directly used by the teacher for adapting the teaching and allocating learning activities based on the students' states, e.g., by providing more personalized and contextualized questions based on their states. For an additional example of the IoT in the domain, see the review by Martin et al. [20].

2.5. Deployment

The IDIVS concept should neither enforce deployment on a specific platform nor enforce deployment solely on a single platform. The idea is that a simple protocol, such as MQTT, can be used in the deployment. However, to be used across many applications,

the deployment should allow for the configuration of which IDIVSs have access to information from which sensors and IDIVSs. Moreover, each IDIVS should have a mechanism, commonly implemented in the form of access control, for ensuring the confidentiality and integrity of internally used data and its input data. This is to ensure the possibility of hiding potentially sensitive, e.g., commercially or privacy-sensitive information, from limited parts of the systems and from services using IDIVS output. The IDIVS can be implemented using push or pull mechanisms. To support dynamic identification of relevant sensor data, we suggest using MQTT topics as a way to communicate relevant sensor properties, including categorizing them by, e.g., physical location being sensed and property type. Each individual IDIVS can then filter out relevant properties by parsing the messages (see Figure 2). In addition, to foster dynamic sensor selection at runtime, a meta-data repository containing the IDIVS information specified in Figure 2 needs to be maintained and updated accordingly. This is also necessary for evaluating the feasibility of directly reusing previously existing IDIVS for new use-cases or for adapting IDIVS to be better suited for deployment to a new physical environment, using techniques such as transfer learning. Providing feature stores and data versioning for the used training datasets is important to ensure continuous integration and delivery.

The current IDIVS reference implementation is implemented as a set of Python scripts, deployed to a cloud computing environment as serverless functions (see Figure 3). In the reference implementation, functions are triggered periodically in order to update their current states, similar to how a sampling sensor would work. An application or a user can also trigger said functions manually to provoke a more recent response. Sensor data is fed to the IDIVS by being buffered in a database before being analysed by the IDIVS. However, the IDIVS architecture allows for direct input of sensor data as well as periodic polling from a database. Sensor data is transported using MQTT, which can be tunnelled over HTTPS. The IDIVS functions poll the stored sensor data using HTTPS. User feedback data is treated as any other data and is stored in the database, and analysed by the IDIVS, e.g., for improving its data fusion model by learning. The MQTT protocol allows a set-up with both a proactive learner (Machine Teacher) and a learner responding to system requests (Active Learning). Resilience and sensor discoverability are achieved by treating missing data as empty datasets, which are disregarded in calculations.

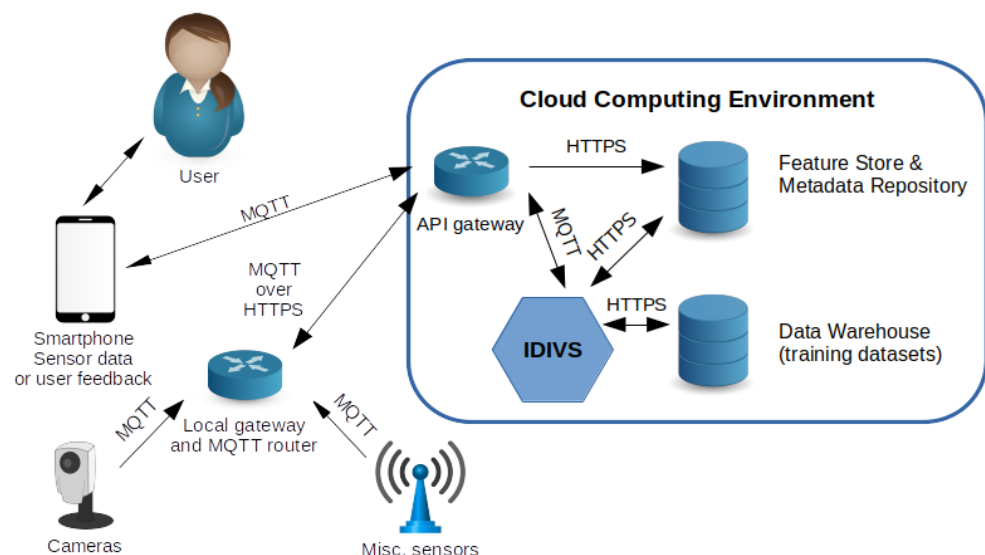


Figure 3. An example of a cloud-based IDIVS deployment.

Thanks to its low overhead, an IDIVS function can also be deployed on, e.g., a gateway or even on a sensor array with an adequate amount of memory and processing power. This allows for a more distributed deployment model, which may fit better in some application

areas. An IDIVS-deployed leaf side can be coupled with announcement functionality to promote dynamic service discoverability within the local environment.

3. Results and Discussion

3.1. Learning Capabilities

To manage the dynamicity of the environment and support efficient set-up of new IDIVSs, we suggest including online machine learning capabilities in IDIVSs. The online learning mechanism is to cater for the situation of the set of available sensors being dynamic, and that the interpretation of environmental states may change over time. We also suggest using data-driven approaches, i.e., black- or grey-box modelling, to facilitate the span of different application areas. If an IDIVS is to use traditional batch learning, the IDIVS is trained offline before it is deployed and thereafter adapted via online training mechanisms. However, an efficient batch learning approach is often not viable due to, e.g., insufficient training data being available. Further, given that the IDIVS concept concerns streamed data, resembling a sensor constantly producing a sensor value, the availability of relevant historical label data may not be natural for some applications. An IDIVS should ideally have learning capabilities that can manage the following:

- Cold start (no or very little initial training data);
- Dynamicity of available sensors;
- Concept drift.

Our primary feature in the IDIVS concept to cope with these aspects is to make use of human users that can provide feedback, or even teach. Moreover, this learning situation can be characterized as being online and incremental.

In previous work, several aspects of the learning element of IDIVS have been explored and illustrated: online learning and cold start setting [21], the dynamic aspect of available sensors [22,23], and the interactive aspect of human-in-the-loop, including both interactive strategies [21] and the human factor [24]. In the following, we highlight the primary findings and connections to the IDIVS' capabilities and design aspects.

3.2. Interactive Learning/Human-In-The-Loop

Machine learning is applied to develop algorithms that can learn from data and improve the knowledge base without human intervention to solve different problems such as pattern recognition, classification, etc. in real time. As mentioned, it is not always possible to achieve high performance without involving humans. Here, interactive learning, i.e., including human-in-the-loop, can be useful to improve the performance of the learning process. One of the main objectives of an IDIVS is to apply feedback from the user in effective ways to improve results, as illustrated in Figure 4. Different types of user feedback can be applied to IDIVSs. The feedback can be explicit, e.g., the user provides input to the IDIVS through a user interface, or implicit, e.g., the user turns on a light switch, providing feedback to the IDIVS that it was too dark.

Furthermore, there might be many alternatives regarding when users provide labels, i.e., different triggers such as the place and time of interaction [25]. A taxonomy of different interactive learning strategies where different types of triggers were identified is presented by Tegen et al. [21]. The taxonomy distinguishes between when the system is more active in the feedback process, as in active learning, and when the user is more proactive, as in machine teaching. The classification of triggers includes the following for active learning: Uncertainty, i.e., the ML acknowledges the uncertainty of the classification; Time, i.e., the system asks the user for a label at a given time interval; and Random, where the user is asked randomly. The classification of triggers for machine teaching includes the following: Error, i.e., the user provides a label when the system is wrong; State change, i.e., the user provides a label when the state of the environment changes; Time, the user provides a label when the user acknowledges a time interval has passed; User factors, i.e., other factors the user may consider for providing a label. We also acknowledge that hybrid versions of the

approaches presented exist, such that the user may be asked to provide labels both based on uncertainty and random.

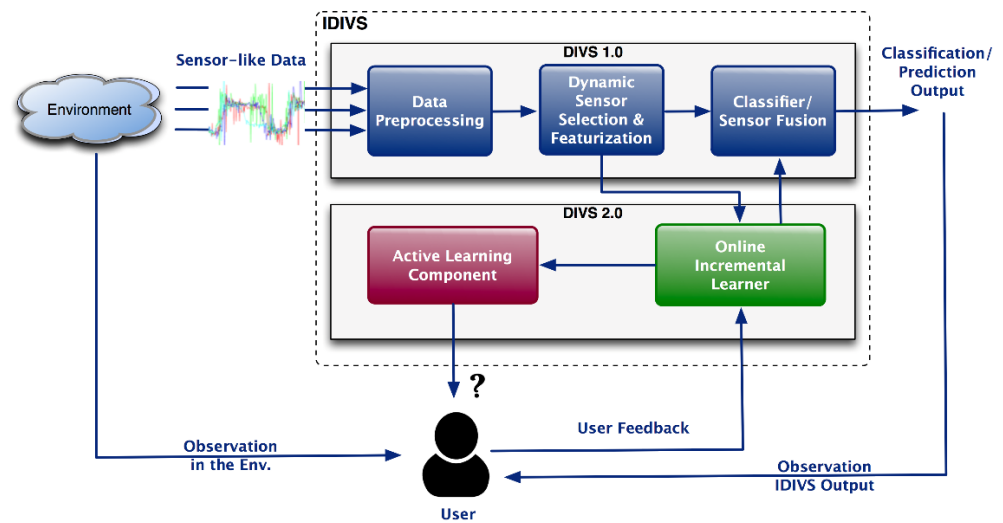


Figure 4. IDIVS employing active learning (an extension of earlier work on DIVS [2]).

Importantly, when designing systems with an IDIVS, we advise considering a service design that allows for both active learning and machine teaching. Hence, this requires that a system of IDIVS and services, such that the user is able to observe the system output as well as the environment, receive requests from the system for feedback, e.g., correct labels, and have efficient means of allowing the user to provide feedback, both on request and pro-actively. The IDIVS suggested capability of supporting self-assessment of the accuracy of its output typically comes into play when an active learning strategy, such as the Uncertainty strategy, is to be used, in which the user is requested to provide labels when the uncertainty is high.

The choice of which particular learning strategy (or hybrid) to use, as introduced above, depends, of course, on the particular application and on the type of user who will be present. Further, the chosen interactive approaches for learning can significantly influence the quality of the classification [22]. As an illustration, please see Figure 5, which clearly indicates that the interactive learning strategy influences performance. The graph shows the average performance of three different learning algorithms, Naïve Bayes (NB), k-Nearest Neighbour (k-NN), and Support Vector Machine (SVM) [22]. In this case, the learning strategy appears to influence the performance much more than the choice of learning algorithms. The labelling budget expresses the number of correct labels provided to the system in relation to the number of data instances streamed. Note, however, that the different learning strategies may imply different amounts of effort for the users. For instance, the Error strategy requires the user to constantly monitor both the system output and the environment, looking for discrepancies.

Although human engagement can aid the training of the machine learning model and increase performance, it also introduces challenges. In active learning, a user is prompted to label the data [9]. However, treating the end user as a passive information oracle has proved problematic because it is not always realistic [21,26]. For instance, in a real-world scenario, a user might not always respond as expected. Tegen et al. [24] explore how the performance of the machine learning classifications is affected if the reliability level of the user is varied. Experiments were conducted where the user did not always respond in accordance with what was expected and where the label provided was not always correct. The interactive learning strategies and machine learning algorithms affected the performance in the experiments, where the combination of Naïve Bayes classifier (compared to SVM and k-NN) and the machine teaching strategy triggered by error resulted in the overall highest performance.

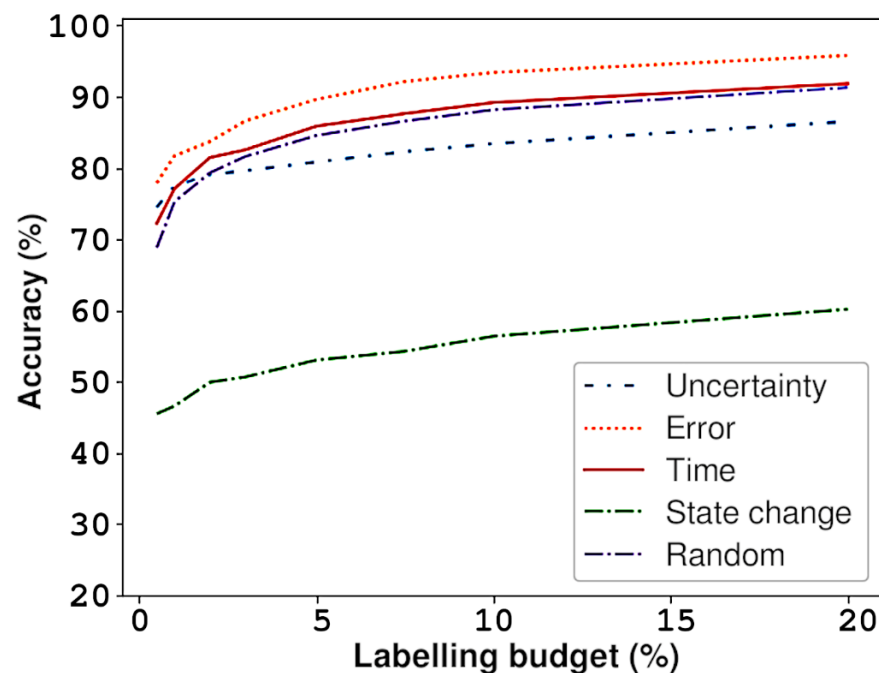


Figure 5. The accumulated average accuracy of different learning algorithms for different labelling budgets and interactive machine learning strategies for the Opportunity dataset. Recompiled results from Tegen et al. [22] into average values of the different learning algorithms (Naïve Bayes (NB), k-Nearest Neighbour (k-NN), and Support Vector Machine, SVM).

There might be many reasons for applying a human-centred design to the interactions between users and a machine learning system. Considering the IDIVS concept, this aspect might be decisive in getting the user to provide relevant and correct feedback. A human-centred approach is often necessary to apply when designing interactions in order to improve the quality of user experiences in this context [27]. It may create a frustrating experience for the end user by neglecting the human ability to experiment and revise, as relevant for the IDIVS concept, thus hindering the user's ability to provide relevant and correct user feedback. We argue that the IDIVS concept allows for a human-centred design by allowing multiple ways to trigger users to provide feedback. Further, it may engage users to provide labels, which is important since labelling is often regarded as tedious and/or costly.

In addition, interactive machine learning models, which may support explanations, feedback, or both, facilitate the interaction of non-expert users with systems for the purpose of improving the learning process, thus increasing system transparency and understandability. This is an approach that can benefit from designing interactions based on user needs and perceptions [28]. Importantly, these systems often rely on user interfaces and interaction modalities to interact with people—visual, audible, tactual, etc. [29]. It has been confirmed that explanations as a dialogue could improve the algorithmic experience in AI systems. For instance, in music recommendation systems, a study [30] confirms that designing experiences that help users understand the underlying algorithmic workings will make recommendations from intelligent systems more usable, thereby providing users with better experiences. It goes beyond the scope of this paper to discuss how to potentially make the IDIVS more explainable. However, we argue that the concept of providing rather minimalistic output, i.e., a virtual sensor value, using appropriate interaction modalities in comparison to a complex information type as output, alleviates the difficulty for a user to understand the output, particularly if the user is also engaged in the teaching of the system.

3.3. Managing Dynamicity and Cold Start

The cold start and dynamicity in activity detection scenarios have been studied, e.g., the cold start setting in Tegen et al. [2], and the dynamic aspect of available sensors [22,23]. The results indicate that the online learning approach is a rather robust approach for handling sensor dynamicity, including being rather straightforward with Naïve Bayes, SVM, and k-NN to adapt the learning to the dynamicity, i.e., the removal and inclusion of sensor input. Obviously, interactive learning is logically sound for coping with cold start scenarios, both in settings with a true cold start (without any training data or trained model existing at the start) [2] and in settings with pre-training that has occurred by, for instance, using transfer learning (see more in Section 3.4). In any case, it requires a user to willingly provide user feedback immediately after the system with IDIVS has started to run. As mentioned, though, such a user needs to be an interactive service user but may also be an IDIVS teacher that takes responsibility for training the IDIVS directly after its deployment.

3.4. Transfer Learning

Transfer learning aims to leverage knowledge acquired from one, or more, so-called source domains (or tasks) to improve performance on a target domain (or task) where data are scarce.

In many real-world applications, the machine learning model trained for a particular setup fails to maintain a high performance level once deployed in a similar, yet different, environment. Consider, for instance, the task of recognizing activities in a smart home environment. A classifier trained based on a dataset provided by one home will generally deliver lower performance when it is directly used in another home. There can be various reasons for this, such as the different distribution of sensors in the home, differences in the plan of the house, different human behaviours, and a varying number of people or sensors in the house, to name a few. Moreover, the problem is exacerbated in situations where a model learned in one environment cannot be directly reused in another environment due to differences in sensing modalities or class labels. Although there could be significant overlap and only slight variation between the two setups, it is often the case that a new model needs to be learned from scratch. The role of transfer learning in this case is to reduce the labelling effort and reuse knowledge acquired from different setups, while, at the same time, maintaining a high performance level.

This aspect is particularly important in the case of IDIVS, where sensing tasks need to be performed in dynamic environments over varying subsets of sensors, which can include new sensing modalities or even augmented target tasks, e.g., new class labels. To this end, we propose a transfer layer that extends the basic functionality of IDIVS, in the sense that it is designed to learn a robust feature representation that can be shared across different related domains. In other words, we aim to learn high-level features that can be used to represent both the source and the target domains. More specifically, we investigate the use of autoencoder-based methods to learn a feature-based transformation where the instances from both the source and target domains are projected to a shared latent space, such that the distances between them are reduced [31]. This approach falls into the category of feature-based transfer learning schemes, as opposed to instance-based methods, where the goal is to reduce the discrepancy between the source and target by reweighting the source samples before training. (For a more detailed review of transfer learning, we refer the reader to Zhuang et al. [10]).

Scenarios can include situations where IDIVSs only have access to (i) unlabelled training data or (ii) few labelled training instances, from the target domain. In the case of the former, the dynamic aspects of IDIVS in particular play a key role. That is, the transfer learning layer incorporates the ability to select those sensors from the target domain, which remain relevant in relation to the sensing task performed by the IDIVS in the source domain(s). For the latter, the interactive feedback mechanism of IDIVS can further be leveraged to select those instances that can contribute most to the domain adaptation phase using active learning (as seen in the Section 3.3).

Practically, suppose we aim to transfer an IDIVS model from a source domain S to a target domain T . In order to align our IDIVS to the new domain, in addition to the general input of the IDIVS introduced in Section 2.3, we also need to provide input to the model, i.e., to the transfer layer of our IDIVS, the differences between the two domains. That is, any additional or missing property types sensed in T in relation to S , as well as any differences in terms of the set of categorical values output by the IDIVS in T in comparison to S . In addition, in future work, we plan to consider user feedback that can present the transfer learning layer with further input regarding how the source and target domains are related, e.g., the positioning of sensors with respect to the plan of the building. Thus, there is a strong emphasis on the human interaction/human-in-the-loop component as part of the transfer learning approach, which can take different forms, depending on the various above-mentioned scenarios. Essentially, this can be viewed as an IDIVS calibration phase where human knowledge is leveraged to increase performance when dealing with a change in domains. In this case, the user assumes the system configurator role according to the categories introduced in Section 2.1. Moreover, input from the source domain concerning the importance of each specific sensor in computing the IDIVS output—i.e., information gain, which is an optional IDIVS parameter—can be leveraged to devise better transfer learning schemes.

3.5. End-User Interface for Automated IDIVS Generation

To efficiently support the creation of new IDIVS and, thereby, new services, we explore interactive mechanisms that can aid the user throughout the process of creating new IDIVS structures, such as the one introduced in Figure 1. To this end, we assume an existing set of physical sensors and a predefined set of existing virtual sensors. The goal is to enable compositional reasoning based on the set of existing sensors in a way that more complex sensing tasks can be automatically performed. The frontend of the proposed approach consists of an interactive way for the user to specify the high-level sensing problem. We resort to a conversational interface, which is an intuitive and user-friendly way in which the user can express the sensing problem and interact with the system, i.e., natural language queries. The backend deploys a deep learning approach that establishes the correspondence between the natural language queries and their virtual sensor representation, i.e., a sequence of computational steps required to implement the desired sensing task. In Mihailescu et al. [32], we demonstrate the approach for a smart city environment, where various types of virtual sensors are scattered and a broad spectrum of sensing tasks can be constructed.

First, we provide an automatic template-based procedure to generate a questions and answers dataset large enough to generalize the association between natural language queries (NLQs) and our proposed virtual sensor functional representation. The functional representation essentially specifies the sequence of functions to be applied to the incoming sensor data. Functions can either implement basic operations, such as arithmetic and logical operators, or correspond to the specific role played by an existing virtual sensor, e.g., car recognition on image data. A detailed discussion on the template-based dataset generation procedure is presented in Mihailescu et al. [32], where we construct a dataset consisting of 48,620 questions with their corresponding functional representation based on a limited number of templates and parameters.

Second, we identify a suitable neural network-based machine learning approach using attention-based gated recurrent units (GRUs), which is able to learn the correspondence between NLQs and virtual sensor representations in a generalized manner. In other words, the model predicts the sequence of computational steps required to implement the desired sensing task expressed in the query. Essentially, this solves the problem introduced in Section 2 of selecting the right set of sensors and virtual sensors, as well as binding them through simple arithmetic and logical operators in order to perform a sensing task that can be expressed by the user through natural language.

As an example, consider the following basic query: Is it colder at Malmö University compared to Malmö Arena? In order to provide an answer, the system constructs an IDIVS that starts by selecting the appropriate type of sensors corresponding to the specified location in the query and returns the result of an arithmetic operation which takes as input the two respective temperature values. This IDIVS generation mechanism can be understood as a configuration phase, where the human user taking the role of the system configurator presents the system with a high-level task specified in natural language, which prompts the system to propose a new IDIVS, constructed based on the pre-existing sensors and virtual sensors in the system. Importantly, the user can inspect the IDIVS structure to determine the validity of the solution (if such a solution exists).

Currently, the mechanism can support four different categories of questions and, since the constructed IDIVS has to adhere to the previously introduced definition of IDIVS, it is mandatory that the question specifies the physical location being sensed. We show that we can efficiently translate natural language questions to functional representations for multi-level IDIVS by constructing a model that can accurately predict answers to the above-mentioned queries, where the answer represents the resulting IDIVS [32]. The proposed GRU-based model achieves a high accuracy of above 99%, when evaluated on the generated dataset and a low word error rate, which is indicative of high performance.

3.6. IDIVS and Information Security

The security goals of any connected device that has sensor capabilities are to protect resources and information from potential attacks [33]. Naturally, this also applies to virtual sensors such as IDIVS and implies that the security requirements of IDIVS should be designed to safeguard the Confidentiality, Integrity, Authenticity and Availability (CIAA) of resources and information. We envisage a threat model consisting of an adversary, with a similar role and capacity to that of an interactive service user and system configurator, but with malicious motivation. The adversary can violate the CIAA security parameters, particularly by injecting malicious input content. Such input can be provided in the form of explicit user feedback, such as labels, when teaching the IDIVS. Consequently, the IDIVS may learn an incorrect model, misclassify data, and detect states that were not planned for at design time. We summarize and exemplify the main attacks targeting IDIVS as follows:

- (i) Adversarial attacks. An adversary may trick the IDIVS ML components, e.g., the Online Incremental Learner, by furnishing malicious input that causes the system to make a false prediction. An example of this is having the adversary intentionally misclassify an activity type.
- (ii) Data poisoning attacks. An adversary may manipulate the input data being used by the IDIVS in a coordinated manner, potentially compromising the entire system. An example of this is having the adversary tamper with the input data to impact the ability of the IDIVS to output correct predictions.
- (iii) Online system manipulation attacks. An adversary may nudge the still-learning IDIVS, i.e., when operating using online learning, in the wrong direction. An example of this is having the adversary reprogram the IDIVS to capture environmental states that were not intended.
- (iv) Transfer learning attacks. An adversary may be able to devise attacks using a pre-trained (full/partial) IDIVS model against a tuned task-specific model. An example of this is having the adversary reverse-engineer the transfer layer of the IDIVS to discover attributes, sensing modalities, and augmented target tasks from the original model.
- (v) Data confidentiality attacks. An adversary may be able to extract confidential or sensitive data that were used for training and teaching the IDIVS. An example of this is having the adversary observe data flows being exchanged between the IDIVS teacher and the learning components.

We posit the importance of providing security-enhancing mechanisms for mitigating the aforementioned attacks, i.e., attacks (i)–(v). For mitigating attack (i), we propose

the use of adversarial training to apply during the IDIVS training process; adversarial training can be conducted by the IDIVS teacher or a derivate of that role. Secure learning algorithms, e.g., algorithms that leverage obfuscation, are suggested as a strategy for mitigating attack (ii). The use of access control models, e.g., capability lists for each IDIVS, is suggested for mitigating attack (iii). To mitigate attack (iv), the transfer layer can be enhanced with functionalities that automatically alter the internal representation of the IDIVS source/target domain models. Finally, for mitigating attack (v), we suggest the use of data encryption. Data encryption is particularly useful when data is being exchanged between the user and the learning components for training the IDIVS.

In addition to the identified attacks and suggested countermeasures, and similar to WSN, IoT, and traditional computer systems, the IDIVS may be prone to other attacks. These attacks may also compromise the CIAA requirements in different ways [34]. Hence, we suggest the deployment of the IDIVS, which offers as a capability (optional or otherwise) a security baseline. This baseline could implement some level of protocol and network security, identity management and access control, and fault tolerance to prevent and detect attacks. We also identified the need for anomaly detection to be integrated into the IDIVS for real-time attack detection. This is important as cyberattacks, particularly those targeting the IoT, including zero-day attacks, are incessantly evolving. These security-enhancing mechanisms may be (partially) provided through the installation of a trusted operating system for enabling the connected devices.

Another suggested mechanism that can help strengthen the security of data and prevent potential attacks, similar to attack (iii), is leveraging technologies like blockchain during incremental training. This mechanism can help keep track of any changes made to the system, including those that are potentially caused by an adversary. Nonetheless, a major challenge in the application of blockchain in such environments is the dynamic and heterogeneous nature of an IoT environment [35]. This makes the implementation of security frameworks based on blockchain more challenging to port to IDIVS. Moreover, it is essential to constantly monitor the incoming/outgoing data flows during the user feedback process to counter possible anomalies that may affect the current IDIVS output. Some examples of anomalies are the deliberate falsification of labels and the injection of malicious content. This will help to distinguish between the available content and what may or may not be malicious during continued learning. Further, machine learning techniques may be considered for the detection of attacks, as suggested and analysed by Waqas et al. [36] and in the more general review by Hussain et al. [37]. Based on the above-mentioned pointers, it is worth highlighting the need for enhancing information security techniques through continued research to protect aspects of the IDIVS in online learning.

3.7. IDIVS Compared to Virtual Sensors

The interactive machine learning and run-time management/adaption of missing sensor data obviously include challenges connected to the possibility of predicting the output from the IDIVS. For a one-shot training approach of data fusion in virtual sensors, it is possible to predict the output given an input, whereas for the interactive learning case, this is not true since the output also depends on future unknown inputs (e.g., labels) provided by users. The complexity of predicting the future output for a given input is also connected to new or disappeared sensors. For the one-shot approach, the output may be rather poor or with a fault indication if sensors disappeared, but rather predictive, whereas for the IDIVS, the output might be accurate enough if the interactive learning has had enough user feedback to adapt to the new situation.

Compared to the one-shot training of the data fusion model, the time aspect of the labels may be more challenging in the interactive ML approach. An example would be if a user provided a label “meeting” as the current activity in an office room at the point when the meeting ended and some people started to leave the room. The sensor values collected at that time may be more representative of the state of people leaving the room or even an empty room if time delays cause the room to become empty when the label is

associated with sensor values in the training. Such a problem can potentially be more easily detected, and data can be curated in a one-shot learning situation compared to an online learning situation. In the online learning case, the problem can, for instance, be alleviated with accurate management of time stamps of sensor data and labels, which, however, may add complexity to the system.

A potential negative aspect of the interactive learning approach is that it typically requires additional computational resources at run-time for the online training compared to a pre-trained data fusion algorithm. This problem may be alleviated by an appropriate choice of ML methods that are resource-efficient with regard to online learning.

4. Conclusions

This paper synthesizes previous DIVS-related results and incorporates them as IDIVS by focusing on mechanisms of machine learning and, in particular, interactive learning. The information properties and capabilities of the IDIVS have been further validated by identifying the requirements of learning mechanisms, including interactive learning and multi-layer learning. In general, the extra requirements of the IDIVS for achieving data fusion based on interactive machine learning are few. However, some types of information, such as property types sensed by sensors or outputted by the IDIVS and information gained, may be highly relevant to achieving transfer learning. Further, the capability to make a self-assessment of the output sensor value may be relevant for some active learning strategies. Moreover, the concept poses some new challenges related to information security. The requirements arising from the consideration of information security aspects have been highlighted, emphasizing the importance of taking user feedback into account in the context of information security. Obviously, tampering with user-supplied labels has a negative impact on the IDIVS output. Thus, we suggest that the user feedback channel be actively monitored to better safeguard the accuracy of the IDIVS. The IDIVS concept seems particularly suitable for interactive learning due to its capability to continuously provide output, which may create engagement for users to provide feedback (most straightforwardly by providing correct labels) to the IDIVS in order to improve its accuracy through machine learning, which is particularly useful in dynamic environments.

Some future directions of research concern the IDIVS concept and the different learning mechanisms and challenges connected to it. One avenue of future research is to make use of other types of feedback for learning than labelled instances, e.g., the relative correctness of the output and implicit feedback from user behaviour, such as turning lighting on or off in the environment. Another avenue concerns the interaction modes connected to user feedback, e.g., using different physical artifacts or other methods of providing user feedback, to increase user engagement and ultimately system accuracy. In addition, exploring identification mechanisms for detecting erroneous user feedback, e.g., due to different interpretations of states, mistakes, or even non-benevolent users, and making use of that information in the learning process is another important topic to research. We believe that approaches such as those based on anomaly detection or voting approaches need adaptations to the IDIVS context.

Other avenues concern the rapid set-up of IDIVS in practice, for instance, regarding the input needed from the installer to quickly achieve high accuracy in the IDIVS. Furthermore, how to efficiently use staged learning/machine teaching with different types of implicit or explicit user input is relevant to address. Finally, one may further consider the challenge of making use of feedback in a multilayer setting of IDIVSs, such that the learning can potentially propagate to other IDIVSs than the one directly receiving a label connected to its output.

Author Contributions: Conceptualization, All; methodology, All; software, J.H., A.T., R.-C.M. and J.B.; validation, All; formal analysis, All; data curation, A.T., J.H., R.-C.M. and J.B.; writing—original draft preparation, All; writing—review and editing, All; project administration, J.A.P.; funding acquisition, P.D. All authors have read and agreed to the published version of the manuscript.

Funding: This research was partially funded by the Knowledge Foundation, grant number 20140035, Internet of Things and People Research Profile.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data sharing not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Mihailescu, R.C.; Persson, J.; Davidsson, P.; Eklund, U. Towards Collaborative Sensing using Dynamic Intelligent Virtual Sensors. In *Intelligent Distributed Computing X. IDC 2016. Studies in Computational Intelligence*; Springer: Cham, Switzerland, 2017; Volume 678. [\[CrossRef\]](#)
2. Tegen, A.; Davidsson, P.; Mihailescu, R.-C.; Persson, J.A. Collaborative Sensing with Interactive Learning Using Dynamic Intelligent Virtual Sensors. *Sensors* **2019**, *19*, 477. [\[CrossRef\]](#) [\[PubMed\]](#)
3. Khan, I.; Belqasmi, F.; Glitho, R.; Crespi, N.; Morrow, M.; Polakos, P. Wireless Sensor Network Virtualization: A Survey. *IEEE Commun. Surv. Tutor.* **2016**, *18*, 553–576. [\[CrossRef\]](#)
4. Liu, L.; Kuo, S.M.; Zhou, M. Virtual Sensing Techniques and Their Applications. In Proceedings of the 2009 International Conference on Networking, Sensing and Control, Okayama, Japan, 26–29 March 2009; pp. 31–36.
5. Islam, M.M.; Hassan, M.M.; Lee, G.-W.; Huh, E.-N. A Survey on Virtualization of Wireless Sensor Networks. *Sensors* **2012**, *12*, 2175–2207. [\[CrossRef\]](#) [\[PubMed\]](#)
6. Kobo, H.I.; Abu-Mahfouz, A.M.; Hancke, G.P. A Survey on Software-Defined Wireless Sensor Networks: Challenges and Design Requirements. *IEEE Access* **2017**, *5*, 1872–1899. [\[CrossRef\]](#)
7. Nguyen, C.; Hoang, D. Software-Defined Virtual Sensors for Provisioning IoT Services on Demand. In Proceedings of the 2020 5th International Conference on Computer and Communication Systems (ICCCS), Shanghai, China, 15–18 May 2020; pp. 796–802.
8. Corsini, P.; Masci, P.; Vecchio, A. Configuration and Tuning of Sensor Network Applications through Virtual Sensors. In Proceedings of the Fourth Annual IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOMW'06), Pisa, Italy, 13–17 March 2006; p. 5.
9. Settles, B. *Active Learning Literature Survey*; University of Wisconsin: Madison, WI, USA, 2010.
10. Zhuang, F.; Qi, Z.; Duan, K.; Xi, D.; Zhu, Y.; Zhu, H.; Xiong, H.; He, Q. A Comprehensive Survey on Transfer Learning. *Proc. IEEE* **2021**, *109*, 43–76. [\[CrossRef\]](#)
11. Kadlec, P.; Gabrys, B.; Strandt, S. Data-Driven Soft Sensors in the Process Industry. *Comput. Chem. Eng.* **2009**, *33*, 795–814. [\[CrossRef\]](#)
12. Aguilera, A.A.; Brena, R.F.; Mayora, O.; Molino-Minero-Re, E.; Trejo, L.A. Virtual Sensors for Optimal Integration of Human Activity Data. *Sensors* **2019**, *19*, 2017. [\[CrossRef\]](#) [\[PubMed\]](#)
13. Aguilera, A.A.; Brena, R.F.; Mayora, O.; Molino-Minero-Re, E.; Trejo, L.A. Multi-Sensor Fusion for Activity Recognition—A Survey. *Sensors* **2019**, *19*, 3808. [\[CrossRef\]](#) [\[PubMed\]](#)
14. Arikumar, K.S.; Tamilarasi, K.; Prathiba, S.B.; Chalapathi, M.V.; Moorthy, R.S.; Kumar, A.D. The Role of Machine Learning in IoT: A Survey. In Proceedings of the 2022 3rd International Conference on Smart Electronics and Communication (ICOSEC), Trichy, India, 20–22 October 2022; pp. 451–457.
15. Asghari, P.; Rahmani, A.M.; Javadi, H.H.S. Internet of Things Applications: A Systematic Review. *Comput. Netw.* **2019**, *148*, 241–261. [\[CrossRef\]](#)
16. Jia, M.; Komeily, A.; Wang, Y.; Srinivasan, R.S. Adopting Internet of Things for the Development of Smart Buildings: A Review of Enabling Technologies and Applications. *Autom. Constr.* **2019**, *101*, 111–126. [\[CrossRef\]](#)
17. Paiva, S.; Ahad, M.A.; Tripathi, G.; Feroz, N.; Casalino, G. Enabling Technologies for Urban Smart Mobility: Recent Trends, Opportunities and Challenges. *Sensors* **2021**, *21*, 2143. [\[CrossRef\]](#) [\[PubMed\]](#)
18. Zantalis, F.; Koulouras, G.; Karabetsos, S.; Kandris, D. A Review of Machine Learning and IoT in Smart Transportation. *Future Internet* **2019**, *11*, 94. [\[CrossRef\]](#)
19. Chen, S.; Jin, X.; Zhang, L.; Wan, J. A Comprehensive Review of IoT Technologies and Applications for Healthcare. In Proceedings of the Advances in Artificial Intelligence and Security, Dublin, Ireland, 19–23 July 2021; Sun, X., Zhang, X., Xia, Z., Bertino, E., Eds.; Springer International Publishing: Cham, Switzerland, 2021; pp. 358–367.
20. Martín, A.C.; Alario-Hoyos, C.; Kloos, C.D. Smart Education: A Review and Future Research Directions. *Proceedings* **2019**, *31*, 57. [\[CrossRef\]](#)
21. Tegen, A.; Davidsson, P.; Persson, J.A. A Taxonomy of Interactive Online Machine Learning Strategies. In *Machine Learning and Knowledge Discovery in Databases. ECML PKDD 2020*; Hutter, F., Kersting, K., Lijffijt, J., Valera, I., Eds.; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2021; Volume 12458. [\[CrossRef\]](#)
22. Tegen, A.; Davidsson, P.; Persson, J. Activity Recognition through Interactive Machine Learning in a Dynamic Sensor Setting. *Pers. Ubiquitous Comput.* **2020**, *2020*, 1–14. [\[CrossRef\]](#)

23. Tegen, A.; Davidsson, P.; Persson, J.A. Interactive Machine Learning for the Internet of Things: A Case Study on Activity Detection. In Proceedings of the 9th International Conference on the Internet of Things; Association for Computing Machinery: Bilbao, Spain, 2019; pp. 1–8. [\[CrossRef\]](#)
24. Tegen, A.; Davidsson, P.; Persson, J.A. The Effects of Reluctant and Fallible Users in Interactive Online Machine Learning. In Proceedings of the Interactive Adaptive Learning 2020, Ghent, Belgium, 14 September 2020; CEUR Workshop Proceedings; CEUR Workshops: Aachen, Germany, 2020; Volume 2660, pp. 55–71.
25. Ghajargar, M.; Persson, J.; Bardzell, J.; Holmberg, L.; Tegen, A. The UX of Interactive Machine Learning. In Proceedings of the 11th Nordic Conference on Human-Computer Interaction: Shaping Experiences, Shaping Society, Tallinn, Estonia, 25–29 October 2020; Association for Computing Machinery: New York, NY, USA, 2020; pp. 1–3.
26. Donmez, P. Proactive Learning: Cost-Sensitive Active Learning with Multiple Imperfect Oracles. In Proceedings of the CIKM08, Napa Valley, CA, USA, 26–30 October 2008.
27. Amershi, S.; Weld, D.; Vorvoreanu, M.; Fournery, A.; Nushi, B.; Collisson, P.; Suh, J.; Iqbal, S.; Bennett, P.N.; Inkpen, K.; et al. Guidelines for Human-AI Interaction. In Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems, Glasgow, UK, 4–9 May 2019; Association for Computing Machinery: Glasgow, UK, 2019; pp. 1–13.
28. Preece, A. Asking ‘Why’ in AI: Explainability of Intelligent Systems—Perspectives and Challenges. *Intell. Syst. Account. Financ. Manag.* **2018**, *25*, 63–72. [\[CrossRef\]](#)
29. Amershi, S.; Cakmak, M.; Knox, W.B.; Kulesza, T. Power to the People: The Role of Humans in Interactive Machine Learning. *AI Mag.* **2014**, *35*, 105–120. [\[CrossRef\]](#)
30. Schröder, A.M.; Ghajargar, M. Unboxing the Algorithm: Designing an Understandable Algorithmic Experience in Music Recommender Systems. In Proceedings of the 15th ACM Conference on Recommender Systems (RecSys 2021), Amsterdam, The Netherlands, 25 September 2021.
31. Mihailescu, R.-C. A Weakly-Supervised Deep Domain Adaptation Method for Multi-Modal Sensor Data. In Proceedings of the 2021 IEEE Global Conference on Artificial Intelligence and Internet of Things (GCAIoT), Dubai, United Arab Emirates, 12–16 December 2021; pp. 45–50.
32. Mihailescu, R.-C.; Kyriakou, G.; Papangelis, A. Natural Language Understanding for Multi-Level Distributed Intelligent Virtual Sensors. *IoT* **2020**, *1*, 494–505. [\[CrossRef\]](#)
33. Li, Y.-X.; Qin, L.; Liang, Q. Research on Wireless Sensor Network Security. In Proceedings of the 2010 International Conference on Computational Intelligence and Security, Nanning, China, 11–14 December 2010; pp. 493–496.
34. Kebande, V.R.; Bugeja, J.; Persson, J.A. Internet of Threats Introspection in Dynamic Intelligent Virtual Sensing. In Proceedings of the 1st Workshop Cyber-Phys. Social Syst. (CPSS), Barcelona, Spain, 17–20 May 2022; pp. 1–8.
35. Ferrag, M.A.; Derdour, M.; Mukherjee, M.; Derhab, A.; Maglaras, L.; Janicke, H. Blockchain Technologies for the Internet of Things: Research Issues and Challenges. *IEEE Internet Things J.* **2019**, *6*, 2188–2204. [\[CrossRef\]](#)
36. Waqas, M.; Kumar, K.; Laghari, A.A.; Saeed, U.; Rind, M.M.; Shaikh, A.A.; Hussain, F.; Rai, A.; Qazi, A.Q. Botnet Attack Detection in Internet of Things Devices over Cloud Environment via Machine Learning. *Concurr. Comput. Pract. Exp.* **2022**, *34*, e6662. [\[CrossRef\]](#)
37. Hussain, F.; Hussain, R.; Hassan, S.A.; Hossain, E. Machine Learning in IoT Security: Current Solutions and Future Challenges. *IEEE Commun. Surv. Tutor.* **2020**, *22*, 1686–1721. [\[CrossRef\]](#)

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.